

CF013444 US/na
Appl No 07/288,038
GAU. 2731

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1999年 4月 5日

出 願 番 号

Application Number:

平成11年特許願第098163号

出 願 人

Applicant (s):

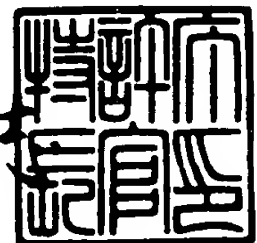
キヤノン株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

1999年 4月30日

特許庁長官
Commissioner,
Patent Office

伴佐山 建志



【書類名】 特許願

【整理番号】 3964012

【提出日】 平成11年 4月 5日

【あて先】 特許庁長官殿

【国際特許分類】 H04L 12/18

【発明の名称】 データ通信システム、装置及び方法並びに記憶媒体

【請求項の数】 37

【発明者】

 【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社
社内

 【氏名】 新井田 光央

【発明者】

 【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社
社内

 【氏名】 波多江 真一

【発明者】

 【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社
社内

 【氏名】 小林 崇史

【発明者】

 【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社
社内

 【氏名】 大西 慎二

【特許出願人】

 【識別番号】 000001007

 【氏名又は名称】 キヤノン株式会社

【代理人】

 【識別番号】 100090273

 【弁理士】

【氏名又は名称】 國分 孝悦

【電話番号】 03-3590-8901

【先の出願に基づく優先権主張】

【出願番号】 平成10年特許願第 97989号

【出願日】 平成10年 4月 9日

【手数料の表示】

【予納台帳番号】 035493

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705348

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ通信システム、装置及び方法並びに記憶媒体

【特許請求の範囲】

【請求項 1】 少なくとも 1 回の非同期通信を用いて 1 つ以上のセグメントからなるオブジェクトデータを転送するソース・ノードと、

前記ソース・ノードから転送されたオブジェクトデータを受信する 1 つ以上のデスティネーション・ノードと、

前記ソース・ノードと前記 1 つ以上のデスティネーション・ノードとの間に論理的な接続関係を設定するコントローラとを備え、

前記コントローラは、複数の異なる通信プロトコルの中から、前記ソース・ノードと前記デスティネーション・ノードとの間で使用される通信プロトコルを選択することを特徴とするデータ通信システム。

【請求項 2】 前記通信プロトコルの 1 つは、ブロードキャスト通信を用いて前記オブジェクトデータを転送する通信プロトコルであることを特徴とする請求項 1 記載のデータ通信システム。

【請求項 3】 前記通信プロトコルの 1 つは、マルチキャスト通信の可能な通信プロトコルであることを特徴とする請求項 1 記載のデータ通信システム。

【請求項 4】 前記通信プロトコルは、Asynchronousブロードキャストトランザクションを用いた通信プロトコルであることを特徴とする請求項 2 又は 3 記載のデータ通信システム。

【請求項 5】 前記通信プロトコルの 1 つは、通信相手を特定する通信方式を用いて前記オブジェクトデータを転送する通信プロトコルであることを特徴とする請求項 1 記載のデータ通信システム。

【請求項 6】 前記通信プロトコルの 1 つは、ユニキャスト通信の可能な通信プロトコルであることを特徴とする請求項 1 記載のデータ通信システム。

【請求項 7】 前記通信プロトコルは、Asynchronousライトトランザクションを用いた通信プロトコルであることを特徴とする請求項 5 又は 6 記載のデータ通信システム。

【請求項 8】 前記ソース・ノードとデスティネーション・ノードとは、自

己の通信能力を記述するレジスタ空間を有することを特徴とする請求項 1～7 のいずれか 1 項に記載のデータ通信システム。

【請求項 9】 前記コントローラは、前記レジスタ空間の内容により前記通信プロトコルを選択することを特徴とする請求項 8 記載のデータ通信システム。

【請求項 10】 前記ソース・ノードとデスティネーション・ノードとは、前記論理的な接続関係に関する情報を記述するレジスタ空間を有することを特徴とする請求項 1～9 のいずれか 1 項に記載のデータ通信システム。

【請求項 11】 前記コントローラは、前記論理的な接続関係を設定する毎に、前記通信プロトコルを選択することを特徴とする請求項 1～10 のいずれか 1 項に記載のデータ通信システム。

【請求項 12】 前記ソース・ノードは、1 セグメント分のデータを少なくとも 1 回の非同期通信を用いて順次転送することを特徴とする請求項 1～11 のいずれか 1 項に記載のデータ通信システム。

【請求項 13】 前記 1 つ以上のデスティネーション・ノードは、前記非同期通信を用いて転送されたデータに対してレスポンスを返すことを特徴とする請求項 1～12 のいずれか 1 項に記載のデータ通信システム。

【請求項 14】 前記コントローラは、前記ソース・ノードと前記 1 つ以上のデスティネーション・ノードとの間に 1 つ以上の論理的な接続関係を設定可能であることを特徴とする請求項 1～13 のいずれか 1 項に記載のデータ通信システム。

【請求項 15】 前記ソース・ノードは、前記 1 つ以上のデスティネーション・ノードのメモリ空間を共通に指定するアドレスを用いて前記オブジェクトデータを書き込むことを特徴とする請求項 1～14 のいずれか 1 項に記載のデータ通信システム。

【請求項 16】 前記通信プロトコルは、Asynchronous Stream packet を用いた通信プロトコルであることを特徴とする請求項 2 又は 3 記載のデータ通信システム。

【請求項 17】 前記データ通信システムは、バス型ネットワークであることを特徴とする請求項 1～16 のいずれか 1 項に記載のデータ通信システム。

【請求項 18】 前記データ通信システムは、IEEE 1394-1995 規格に準拠したネットワークであることを特徴とする請求項 1～17 のいずれか 1 項に記載のデータ通信システム。

【請求項 19】 前記オブジェクトデータは、静止画像データ、グラフィックスデータ、テキストデータ、ファイルデータ、プログラムデータの少なくとも 1 つであることを特徴とする請求項 1～18 のいずれか 1 項に記載のデータ通信システム。

【請求項 20】 ソース・ノードと 1 つ以上のデスティネーション・ノードとの間に論理的な接続関係を設定するステップと、

複数の異なる通信プロトコルの中から、前記ソース・ノードと前記デスティネーション・ノードとの間で使用される通信プロトコルを選択するステップと、

1 つ以上のセグメントからなるオブジェクトデータを少なくとも 1 回の非同期通信を用いて前記デスティネーション・ノードに転送するステップとを有することを特徴とするデータ通信方法。

【請求項 21】 1 つ以上のセグメントからなるオブジェクトデータを少なくとも 1 つの通信パケットにパケット化する手段と、

1 つ以上のデスティネーション・ノードとの間に設定された論理的な接続関係と複数の異なる通信プロトコルの 1 つとを用いて、前記通信パケットを非同期転送する手段とを備えることを特徴とするデータ通信装置。

【請求項 22】 1 つ以上のセグメントからなるオブジェクトデータを少なくとも 1 つの通信パケットにパケット化するステップと、

1 つ以上のデスティネーション・ノードとの間に設定された論理的な接続関係と複数の異なる通信プロトコルの 1 つとを用いて、前記通信パケットを非同期転送するステップとを有することを特徴とするデータ通信方法。

【請求項 23】 ソース・ノードとの間に設定された論理的な接続関係と複数の異なる通信プロトコルの 1 つとを用いて非同期転送された少なくとも 1 つの通信パケットを受信する手段と、

前記通信パケットに含まれるデータを該通信パケットの指定するメモリ空間に書き込む手段とを備えることを特徴とするデータ通信装置。

【請求項 24】 ソース・ノードとの間に設定された論理的な接続関係と複数の異なる通信プロトコルの 1 つとを用いて非同期転送された少なくとも 1 つの通信パケットを受信するステップと、

前記通信パケットに含まれるデータを該通信パケットの指定するメモリ空間に書き込むステップとを有することを特徴とするデータ通信方法。

【請求項 25】 ソース・ノードと 1 つ以上のデスティネーション・ノードとの間に論理的な接続関係を設定する手段と、

前記論理的な接続関係において使用可能な通信プロトコルを、複数の異なる通信プロトコルの中から選択する手段とを備えることを特徴とするデータ通信装置。

【請求項 26】 ソース・ノードと 1 つ以上のデスティネーション・ノードとの間に論理的な接続関係を設定するステップと、

前記論理的な接続関係において使用可能な通信プロトコルを、複数の異なる通信プロトコルの中から選択するステップとを有することを特徴とするデータ通信方法。

【請求項 27】 情報データを送信するソース・ノードと、該情報データを受信するデスティネーション・ノードとを含むデータ通信システムにおいて、

1 つ以上の特定された転送先或いは不特定多数の転送先に応じて異なる複数の通信プロトコルの何れかを選択し、上記情報データを転送することを特徴とするデータ通信システム。

【請求項 28】 上記異なる複数の通信プロトコルの夫々に関する情報は、上記ソース・ノード及び上記デスティネーション・ノードの具備する記憶手段に記憶されていることを特徴とする請求項 27 記載のデータ通信システム。

【請求項 29】 上記異なる複数の通信プロトコルの 1 つは、IEEE 1394-1995 規格に準拠した Asynchronous ライトランザクションを用いた通信プロトコルであることを特徴とする請求項 27 又は 28 記載のデータ通信システム。

【請求項 30】 上記異なる複数の通信プロトコルの 1 つは、IEEE 1394-1995 規格に準拠した Asynchronous ブロードキャストトラ

ンザクションを用いた通信プロトコルであることを特徴とする請求項 27～29 のいずれか 1 項に記載のデータ通信システム。

【請求項 31】 上記異なる複数の通信プロトコルの 1 つは、IEEE1394-1995 規格に準拠した Asynchronous Streams を用いた通信プロトコルであることを特徴とする請求項 27～29 のいずれか 1 項に記載のデータ通信システム。

【請求項 32】 情報データを送信するソース・ノードと、該情報データを受信するデスティネーション・ノードとを含むデータ通信システムにおいて、

上記ソース・ノードと上記デスティネーション・ノードとの間の接続を設定する迄の動作を共用する複数の通信プロトコルの何れかを選択し、前記情報データを転送することを特徴とするデータ通信システム。

【請求項 33】 情報データを送信するソース・ノードと、該情報データを受信するデスティネーション・ノードとを含むデータ通信システムに適用可能なデータ通信方法において、

1 つ以上の特定された転送先或いは不特定多数の転送先に応じて異なる複数の通信プロトコルの何れかを選択し、上記情報データを転送することを特徴とするデータ通信方法。

【請求項 34】 情報データを送信するソース・ノードと、該情報データを受信するデスティネーション・ノードとを含むデータ通信方法において、

上記ソース・ノードと上記デスティネーション・ノードとの間の接続を設定する迄の動作を共用する複数の通信プロトコルの何れかを選択し、前記情報データを転送することを特徴とするデータ通信方法。

【請求項 35】 情報データを送信するソース・ノードと、該情報データを受信するデスティネーション・ノードとを含むデータ通信システムに適用可能なデータ通信装置において、

1 つ以上の特定された転送先或いは不特定多数の転送先に応じて異なる複数の通信プロトコルの何れかを選択する選択手段と、

上記選択手段により選択された通信プロトコルに基づいて情報データの転送を行う転送手段とを具備することを特徴とするデータ通信装置。

【請求項 36】 請求項 20、22、24、26、33、34 のいずれか 1 項に記載のデータ通信方法の手順をコンピュータに実行させるためのプログラムを格納したことを特徴とする記憶媒体。

【請求項 37】 請求項 21、23、25、35 のいずれか 1 項に記載の各手段としてコンピュータを機能させるためのプログラムを格納したことを特徴とする記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明はデータ通信システム、装置及び方法並びに記憶媒体に関し、特に、情報データ（画像データを含む）とコマンドデータとを混在させて高速に通信するネットワークとそのネットワークに適用可能な通信プロトコルに関するものである。

【0002】

【従来の技術】

従来、パーソナルコンピュータ（以下、PC）の周辺機器の中で、最も利用頻度が高かったのはハードディスクやプリンタであった。これらの周辺機器は、専用の入出力インタフェース或いは SCSI（small computer system interface）インタフェース等の汎用性のあるデジタルインタフェースによって PC と接続されていた。

【0003】

一方、近年、デジタルカメラ、デジタルビデオカメラ等の AV（Audio/Visual）機器も PC の周辺装置の 1 つとして注目を浴びている。これらの AV（Audio/Visual）機器も専用インタフェースを介して PC と接続されていた。

【0004】

ところが、従来の周辺機器の有するデジタルインタフェースと AV 機器の有するデジタルインタフェースとは互換性がなく、それらを直接接続することはできなかった。そのため、例えば、AV 機器が周辺機器に、例えば、静止画像を通信したい場合、必ず PC を介す必要があった。

【0005】

又、従来の専用インタフェースやSCSIインタフェースでは、特にAV機器の有する静止画像や動画画像のような大容量のデータを扱う場合において、データ転送レートが低い、パラレル通信のため通信ケーブルが太い、接続できる周辺機器の数と種類が少ない、接続方式に制限がある、リアルタイムなデータ転送が行えない等の多くの問題があった。

【0006】

このような問題点を解決する次世代の高速、高性能デジタルインタフェースの1つとして、IEEE (The Institute of Electrical and Electronics Engineers, Inc.) 1394-1995規格が知られている。

【0007】

IEEE 1394-1995規格に準拠したデジタルインタフェース（以下、1394インタフェース）には、次のような特徴がある。

- (1) データ転送速度が高速である。
- (2) リアルタイムなデータ転送方式（即ち、Isochronous転送方式）とAsynchronous転送方式をサポートしている。
- (3) 自由度の高い接続構成（トポロジ）を構築できる。
- (4) プラグ・アンド・プレイ機能と活線挿抜機能をサポートしている。

【0008】

【発明が解決しようとする課題】

しかしながら、IEEE 1394-1995規格では、コネクタの物理的、電氣的な構成、最も基本的な2つのデータ転送方式等については定義しているが、どのような種類のデータを、どのようなデータ・フォーマットで、どのような通信プロトコルに基づいて送受信するのかについては定義していなかった。

【0009】

又、IEEE 1394-1995規格のIsochronous転送方式では、送出パケットに対する応答が規定されていないため、各Isochronousパケットが確実に受信されたかは保証されない。従って、連続性のある複数のデータを確実に転送したい場合、或いは1つのファイルデータを複数のデータに

分割して確実に転送したい場合、Isochronous 転送方式を使用することはできなかった。

【0010】

又、IEEE 1394-1995 規格の Isochronous 転送方式では、転送帯域に空きがある場合でも、通信の総数が 64 個に制限される。このため、少ない転送帯域で数多くの通信を行いたい場合、Isochronous 転送方式を使用することはできなかった。

【0011】

又、IEEE 1394-1995 規格では、ノードの電源の ON/OFF、ノードの接続/切り離し等に応じてバスリセットが生じた場合、データの転送を中断しなければならない。ところが、IEEE 1394-1995 規格では、バスリセットや伝送時のエラーによってデータ転送が中断した場合、どのような内容のデータを失ったのかについては知ることができなかった。更に、一度中断した転送を復帰するためには、非常に煩雑な通信手順を踏む必要があった。ここで、バスリセットとは、新たなトポロジの認識と各ノードに割り当てられるアドレス（ノード ID）の設定とを自動的に行う機能である。この機能により、IEEE 1394-1995 規格では、プラグ・アンド・プレイ機能と活線挿抜機能とを提供することができる。

【0012】

又、IEEE 1394-1995 規格に準拠した通信システムにおいて、リアルタイム性は必要とされないが、信頼性が要求される比較的数据量の多いオブジェクトデータ（例えば、静止画像データ、グラフィックスデータ、テキストデータ、ファイルデータ、プログラムデータ等）を、1 以上のセグメントに分割し、各セグメントを 1 つ以上の Asynchronous 転送を用いて順次転送するための通信プロトコルは具体的に提案されていなかった。

【0013】

特に、IEEE 1394-1995 規格に準拠した通信システムにおいて、複数の機器間のデータ通信を、Asynchronous 転送方式によるブロードキャスト通信を用いて実現するための通信プロトコルも具体的に提案されてい

かった。

【0014】

又、このような通信プロトコルが複数個提案された場合、通信相手の使用する通信プロトコルを自動的に判別し、使い分けるようにできることが望ましい。しかしながら、複数の異なる通信プロトコルを使い分けるための仕組みについても具体的には何も提案されていなかった。

【0015】

本発明は上述の如き問題を解決することを目的とする。

本発明の他の目的データ通信システム、装置及び方法並びに記憶媒体において、リアルタイム性の必要としないオブジェクトデータを確実に転送することのできる技術を提供する。

又、本発明の他の目的は、データ通信システム、装置及び方法並びに記憶媒体において、複数の異なる通信プロトコルが存在する場合でも、使用可能な通信プロトコルを適応的に選択することのできる技術を提供する。

【0016】

【課題を解決するための手段】

本発明のデータ通信システムは、少なくとも1回の非同期通信を用いて1つ以上のセグメントからなるオブジェクトデータを転送するソース・ノードと、前記ソース・ノードから転送されたオブジェクトデータを受信する1つ以上のデスティネーション・ノードと、前記ソース・ノードと前記1つ以上のデスティネーション・ノードとの間に論理的な接続関係を設定するコントローラとを備え、前記コントローラは、複数の異なる通信プロトコルの中から、前記ソース・ノードと前記デスティネーション・ノードとの間で使用される通信プロトコルを選択することを特徴とする。

【0017】

本発明のデータ通信方法は、ソース・ノードと1つ以上のデスティネーション・ノードとの間に論理的な接続関係を設定するステップと、複数の異なる通信プロトコルの中から、前記ソース・ノードと前記デスティネーション・ノードとの間で使用される通信プロトコルを選択するステップと、1つ以上のセグメントか

らなるオブジェクトデータを少なくとも1回の非同期通信を用いて前記デスティネーション・ノードに転送するステップとを有することを特徴とする。

【0018】

本発明のデータ通信装置は、1つ以上のセグメントからなるオブジェクトデータを少なくとも1つの通信パケットにパケットサイズする手段と、1つ以上のデスティネーション・ノードとの間に設定された論理的な接続関係と複数の異なる通信プロトコルの1つとを用いて、前記通信パケットを非同期転送する手段とを備えることを特徴とする。

【0019】

本発明のデータ通信方法は、1つ以上のセグメントからなるオブジェクトデータを少なくとも1つの通信パケットにパケットサイズするステップと、1つ以上のデスティネーション・ノードとの間に設定された論理的な接続関係と複数の異なる通信プロトコルの1つとを用いて、前記通信パケットを非同期転送するステップとを有することを特徴とする。

【0020】

本発明のデータ通信装置は、ソース・ノードとの間に設定された論理的な接続関係と複数の異なる通信プロトコルの1つとを用いて非同期転送された少なくとも1つの通信パケットを受信する手段と、前記通信パケットに含まれるデータを該通信パケットの指定するメモリ空間に書き込む手段とを備えることを特徴とする。

【0021】

本発明のデータ通信方法は、ソース・ノードとの間に設定された論理的な接続関係と複数の異なる通信プロトコルの1つとを用いて非同期転送された少なくとも1つの通信パケットを受信するステップと、前記通信パケットに含まれるデータを該通信パケットの指定するメモリ空間に書き込むステップとを有することを特徴とする。

【0022】

本発明のデータ通信装置は、ソース・ノードと1つ以上のデスティネーション・ノードとの間に論理的な接続関係を設定する手段と、前記論理的な接続関係に

において使用可能な通信プロトコルを、複数の異なる通信プロトコルの中から選択する手段とを備えることを特徴とする。

【0023】

本発明のデータ通信方法は、ソース・ノードと1つ以上のデスティネーション・ノードとの間に論理的な接続関係を設定するステップと、前記論理的な接続関係において使用可能な通信プロトコルを、複数の異なる通信プロトコルの中から選択するステップとを有することを特徴とする。

【0024】

本発明のデータ通信システムは、情報データを送信するソース・ノードと、該情報データを受信するデスティネーション・ノードとを含むデータ通信システムにおいて、1つ以上の特定された転送先或いは不特定多数の転送先に応じて異なる複数の通信プロトコルの何れかを選択し、上記情報データを転送することを特徴とする。

【0025】

本発明のデータ通信システムは、情報データを送信するソース・ノードと、該情報データを受信するデスティネーション・ノードとを含むデータ通信システムにおいて、上記ソース・ノードと上記デスティネーション・ノードとの間の接続を設定する迄の動作を共用する複数の通信プロトコルの何れかを選択し、前記情報データを転送することを特徴とする。

【0026】

本発明のデータ通信方法は、情報データを送信するソース・ノードと、該情報データを受信するデスティネーション・ノードとを含むデータ通信システムに適用可能なデータ通信方法において、1つ以上の特定された転送先或いは不特定多数の転送先に応じて異なる複数の通信プロトコルの何れかを選択し、上記情報データを転送することを特徴とする。

【0027】

本発明のデータ通信方法は、情報データを送信するソース・ノードと、該情報データを受信するデスティネーション・ノードとを含むデータ通信方法において、上記ソース・ノードと上記デスティネーション・ノードとの間の接続を設定

する迄の動作を共用する複数の通信プロトコルの何れかを選択し、前記情報データを転送することを特徴とする。

【0028】

本発明のデータ通信装置において、情報データを送信するソース・ノードと、該情報データを受信するデスティネーション・ノードとを含むデータ通信システムに適用可能なデータ通信装置において、1つ以上の特定された転送先或いは不特定多数の転送先に応じて異なる複数の通信プロトコルの何れかを選択する選択手段と、上記選択手段により選択された通信プロトコルに基づいて情報データの転送を行う転送手段とを具備することを特徴とする。

【0029】

【発明の実施の形態】

以下、本発明の好適な実施例について図面を参照して詳細に説明する。

図1は、データ通信システムの構成の一例を示す図である。本実施例のデータ通信システムは、図1に示すように、コンピュータ10、カメラ一体型デジタルビデオレコーダ28、プリンタ60により構成されている。

【0030】

まず、コンピュータ10の構成について説明する。12はコンピュータ10の動作を制御する演算処理装置(MPU)である。14はIEEE1394-1995規格に準拠した機能と本実施例において規定する通信プロトコルに関する機能とを有する1394インタフェースである。16はキーボード、マウスなどからなる操作部である。18は圧縮符号化されたデジタルデータ(動画像データ、静止画像データ、音声データ等)を復号するデコーダである。20はCRTディスプレイや液晶パネルなどの表示装置からなる表示部(ディスプレイ)である。22は各種のデジタルデータ(動画像データ、静止画像データ、音声データ、グラフィックスデータ、テキストデータ、プログラムデータ等)を記録するハードディスク(HD)である。24は内部メモリである。26はPCIバスなどのコンピュータ10内部の各処理部を相互に接続する内部バスである。

【0031】

次に、カメラ一体型デジタルビデオレコーダ(以下、DVCRと称する)28

の構成について説明する。30は被写体の光学像を電気信号に変換し、その電気信号をに変換する撮像部(opt)である。32はアナログ-デジタル(A/D)変換器である。34はデジタル化された動画像、静止画像を所定のフォーマットのデジタル画像データに変換する画像処理部である。36は圧縮/伸長処理部であり、圧縮符号化されたデジタルデータ(動画像データ、静止画像データ、音声データ等)を復号する機能と、デジタル画像データを高能率符号化する(例えば、MPEG方式やDV方式のように、所定の画像単位に直交変換後、量子化し、可変長符号化する)機能とを有する。38は高能率符号化されたデジタル画像データを一時的に格納するメモリである。40は高能率符号化されていないデジタル画像データを一時的に格納するメモリである。42はデータセレクタである。44はIEEE1394-1995規格に準拠した機能と本実施例において規定する通信プロトコルに関する機能とを有する1394インタフェースである。46、48はメモリ38とメモリ40の書き込みと読み出しとを制御するメモリ制御部である。50はDVCR28の動作を制御する制御部(システムコントローラ)であり、マイクロコンピュータを有する。52はリモコンや操作パネル等からなる操作部である。54は電子ビューファインダ(EVF)である。56はD/A変換器である。58は磁気テープ、磁気ディスク、光磁気ディスク等の記録媒体からなる記録再生部で、各種のデジタルデータ(動画像データ、静止画像データ、音声データ等)を記録再生する。

【0032】

次に、プリンタ60の構成について説明する。62はIEEE1394-1995規格に準拠した機能と本実施例において規定する通信プロトコルに関する機能とを有する1394インタフェースである。64はデータセレクタである。66は操作ボタンやタッチパネル等からなる操作部である。68はプリンタ60の動作を制御するプリンタコントローラである。70はデコーダである。72は内部メモリである。74は1394インタフェースを介して受信された静止画像データ、テキストデータ、グラフィックスデータ等処理する画像処理部である。76はドライバである。78はプリンタヘッドである。

【0033】

図1に示すように、コンピュータ10、DVCR28及びプリンタ60の各通信装置（以下、ノードと称する）は、1394インタフェース14、44、62を介して相互に接続されている（以下、1394インタフェースによって構成されたネットワークを1394シリアルバスと称する）。各ノードは、所定の通信プロトコルを定義することによって、各種のオブジェクトデータ（例えば、動画像データ、静止画像データ、音声データ、グラフィックスデータ、テキストデータ、プログラムデータ等）の授受、コマンドデータによる遠隔操作が可能となる。

【0034】

次に、図1を用いて本実施例の通信システムを構成する各ノードの動作について説明する。

まず、コンピュータ10を構成する各処理部の機能と動作について説明する。本実施例においてコンピュータ10は、例えば、DVCR28とプリンタ60との間における画像データの送受信を制御するコントローラ、或いはDVCR28やプリンタ60を遠隔操作するコントローラとして動作する。

【0035】

MPU12は、ハードディスク22に記録されているソフトウェアを実行するとともに、様々なデータを内部メモリ24に移動させる。又、MPU12は、内部バス26によって接続されている各処理部の調停動作なども合わせて行なう。

【0036】

1394インタフェース14は、1394シリアルバス上に転送された画像データを受信するとともに、ハードディスク22や内部メモリ24に記録されている画像データを1394シリアルバス上に送信することができる。又、1394インタフェース14は、1394シリアルバス上の他のノードを遠隔操作するためのコマンドデータを送信することも可能である。更に、1394インタフェース14は、1394シリアルバスを介して転送された信号を他のノードに転送する機能も有している。

【0037】

ユーザは、操作部16を介して所望のソフトウェアを選択し、MPU12にハ

ードディスク 22 に記録されているソフトウェアを実行させる。ここで、このソフトウェアに関する情報は、表示部 20 によってユーザに提示される。デコーダ 18 は、このソフトウェアに基づいて、1394 シリアルバス上から受信した画像データをデコードする。デコードされた画像データは、表示部 20 によってユーザに提示される。

【0038】

次に、DVCR 28 を構成する各処理部の機能と動作について説明する。本実施例において DVCR 28 は、本実施例の通信プロトコルに基づいて画像データを転送する画像送信装置（ソース・ノード）として動作する。

【0039】

撮像部 30 は、被写体の光学像を輝度信号（Y）と色差信号（C）とからなる電気信号に変換し、その電気信号を A/D 変換器 60 に供給する。A/D 変換器 32 は、その電気信号をデジタル化する。

【0040】

画像処理部 34 は、デジタル化された輝度信号と色差信号とに対して所定の画像処理を施すと共に、それらを多重化する。圧縮／伸長処理部 36 は、デジタル化された輝度信号と色差信号のデータ量を圧縮する。ここで、圧縮／伸長処理部 36 は、独立した圧縮処理回路を用いて輝度信号と色差信号と並列に処理してもよい。又、共通の圧縮処理回路を用いてそれらを時分割に処理してもよい。又、圧縮／伸長処理部 36 では、伝送路誤りに強くするために、圧縮画像データに対してシャフリング処理を施す。これにより、連続的な符号誤り（即ち、バーストエラー）を、修整や補間の行いやすい離散的な誤り（即ち、ランダムエラー）に変換することができる。ここで、画像の画面内の粗密による情報量の偏りを均一化したい場合、圧縮処理の前に本処理工程を持ってくると、ランレングス等の可変長符号化を用いた場合の都合が良い。

【0041】

圧縮／伸長処理部 36 では、圧縮画像データに対して、シャフリングを復元するためのデータ識別情報（ID）を付加する。圧縮／伸長処理部 36 は、記録再生時の誤りを低減する為に、圧縮画像データに対してエラー訂正符号（ECC）を

付加する。

【0042】

圧縮／伸長処理部 36 にて圧縮された画像データは、メモリ 38 と記録再生部 58 とに供給される。記録再生部 58 は、ID や ECC の付加された圧縮画像データを磁気テープ等の記録媒体に記録する。ここで、圧縮画像データは、音声データとは異なる独立の記録エリアに記録される。

【0043】

一方、画像処理部 34 から D/A 変換器 56 へ供給された画像データは、D/A 変換される。EVF 54 は、D/A 変換器 56 から供給されたアナログ画像信号を表示する。又、画像処理部 34 にて処理された画像データは、メモリ 40 にも供給される。ここで、メモリ 40 には、非圧縮の画像データが格納される。

【0044】

データセクタ 42 は、ユーザの指示に基づいてメモリ 38 或いはメモリ 40 を選択し、圧縮画像データ或いは非圧縮画像データを 1394 インタフェース 44 に供給する。又、データセクタ 42 は、1394 インタフェース 44 から供給された画像データをメモリ 38 或いはメモリ 40 に供給する。

【0045】

1394 インタフェース 44 は、後述する本実施例の通信プロトコルに基づいて、圧縮画像データ或いは非圧縮画像データを転送する。又、1394 インタフェース 44 は、1394 シリアルバスを介して、DVCR 28 を制御するための制御コマンドを受信する。受信された制御コマンドは、データセクタ 42 を介して、制御部 50 に供給される。1394 インタフェース 44 は、上記の制御コマンドに対するレスポンスを返送する。

【0046】

次に、プリンタ 60 を構成する各処理部の機能と動作について説明する。本実施例においてプリンタ 60 は、本実施例の通信プロトコルに基づいて転送された画像データを受信して印刷する画像受信装置（デスティネーション・ノード）として動作する。

【0047】

1394 インタフェース 62 は、1394 シリアルバスを介して転送された画像データや制御コマンドを受信する。又、1394 インタフェース 62 は、制御コマンドに対するレスポンスを送信する。

【0048】

受信された画像データは、データセレクタ 64 を介して、デコーダ 70 に供給される。デコーダ 70 は、該画像データをデコードし、その結果を画像処理部 74 に出力する。画像処理部 74 は、デコードされた画像データをメモリ 72 に一時的に記憶する。

【0049】

又、画像処理部 74 は、メモリ 72 に一時的に記憶された画像データを印刷用のデータに変換し、それをプリンタヘッド 78 に供給する。プリンタヘッド 78 は、プリンタコントローラ 68 の制御に基づき、印刷を実行する。

【0050】

一方、受信された制御コマンドは、データセレクタ 64 を介して、プリンタコントローラ 68 に入力される。プリンタコントローラ 68 は、該制御データに基づいて印刷に関する様々な制御を行なう。例えば、ドライバ 76 による紙送り、プリンタヘッド 78 の位置等を制御する。

【0051】

次に、図 2 を用いて本実施例の 1394 インタフェース 14、44、62 の構成について詳細に説明する。

1394 インタフェースは、機能的に複数のレイヤ（階層）から構成されている。図 2 において、1394 インタフェースは、IEEE 1394-1995 規格に準拠した通信ケーブル 201 を介して他のノードの 1394 インタフェースと接続される。又、1394 インタフェースは、1 つ以上の通信ポート 202 を有し、各通信ポート 802 はハードウェア部に含まれるフィジカル・レイヤ 203 と接続される。

【0052】

図 2 において、ハードウェア部は、フィジカル・レイヤ 203 とリンク・レイヤ 204 とから構成されている。フィジカル・レイヤ 203 は、他のノードとの

物理的、電氣的なインタフェース、バスリセットの検出とそれに伴う処理、入出力信号の符号化／復号化、バス使用権の調停等を行う。又、リンク・レイヤ 204 は、通信パケットの生成、各種の通信パケットの送受信、サイクルタイマの制御等を行なう。更に、リンク・レイヤ 204 は、後述する第 1、第 2 の通信プロトコルで使用するデータパケットの生成及び送受信の機能を提供する。

【0053】

又、図 2 において、ファームウェア部は、トランザクション・レイヤ 205 とシリアル・バス・マネージメント 206 とを含んでいる。トランザクション・レイヤ 205 は、Asynchronous 転送方式を管理し、各種のトランザクション（リード、ライト、ロック）を提供する。更に、トランザクション・レイヤ 205 は、後述する第 1、第 2 の通信プロトコルの機能を提供する。シリアル・バス・マネージメント 206 は、後述する IEEE 1212 CSR 規格に基づいて、自ノードの制御、自ノードの接続状態の管理、自ノードの ID 情報の管理、シリアルバスネットワークの資源管理を行う機能を提供する。

【0054】

図 2 に示すハードウェア部及びファームウェア部が実質的に 1394 インタフェースを構成するものであり、それらの基本構成は IEEE 1394-1995 規格により規定されている。

【0055】

又、ソフトウェア部に含まれるアプリケーション・レイヤ 207 は、使用するアプリケーションソフトによって異なり、どのようなオブジェクトデータを第 1、第 2 の通信プロトコルを用いてどのように転送するかを制御する。

【0056】

後述する本実施例の第 1、第 2 の通信プロトコルは、1394 インタフェースを構成するハードウェア部及びファームウェア部の機能を拡張するものであり、ソフトウェア部に対して新規な転送手順を提供するものである。

【0057】

次に、図 3 を用いて、本実施例の第 1、第 2 の通信プロトコルの基本構成について説明する。本実施例において、第 1 の通信プロトコルは、マルチキャスト通

信を実現する通信プロトコルである。又、第2の通信プロトコルは、ユニキャスト通信を実現する通信プロトコルである。

【0058】

図3において、300はコントローラ、302はソース・ノード、304は n ($n \geq 1$) 個のデスティネーション・ノード、306はソース・ノードの有するサブユニット (subunit)、308は動画像データ、静止画像データ、グラフィックスデータ、テキストデータ、ファイルデータ、プログラムデータ等のオブジェクトデータ (object) である。

【0059】

310はデスティネーション・ノード304内部にある第1のメモリ空間であり、所定のデスティネーション・オフセット (destination_offset_0) により指定される。312はソース・ノード302とデスティネーション・ノード304との間の論理的な接続関係 (以下、コネクションと称する) を示す第1のコネクションである。ここで、デスティネーション・オフセットとは、 n 個のデスティネーション・ノード304の有するメモリ空間を共通に指定するアドレスである。

【0060】

314はデスティネーション・ノード304内部にある第 n のメモリ空間であり、所定のデスティネーション・オフセット (destination_offset_n) により指定される。316はソース・ノード302とデスティネーション・ノード304との間のコネクションを示す第 n のコネクションである。

【0061】

本実施例において、各ノードは、第1のメモリ空間310～第 n のメモリ空間314を、IEEE1212 CSR (Control and Status Register Architecture) 規格 (又は、ISO/IEC 13213:1994 規格) に準拠した64ビットのアドレス空間により管理している。IEEE1212 CSR規格とは、シリアルバス向けの制御、管理、アドレス割り振りを規定した規格である。

【0062】

図4は、各ノードの有するアドレス空間について説明する図である。図4（a）は、各ノードのメモリ空間をアドレッシングする64ビットのアドレス空間である。64ビットアドレスの上位16ビットは、あるバス内のあるノードを指定するために使用される。又、図4（b）は、図4（a）に示すアドレス空間の一部であり、例えば、上位16ビットで指定されたノードのもつあるアドレス空間を指定するために使用される。

【0063】

図3に示す第1のメモリ空間310～第nのメモリ空間314は、図4（b）に示す48ビットのアドレス（以下、デスティネーション・オフセットと称する）により指定される。図4（b）において、例えば、 00000000000000_{16} ～ $00000000003FF_{16}$ は予約された領域であり、実際にオブジェクトデータ308の書き込まれる領域は、アドレスの下位48ビットがFFFFF0000400₁₆以降となる領域である。

【0064】

図3において、ソース・ノード302とは、第1或いは第2の通信プロトコルに従ってオブジェクトデータ308を転送する機能をもつノードであり、デスティネーション・ノード304とは、ソース・ノード302から転送されたオブジェクトデータ308を受信する機能をもつノードである。又、コントローラ300とは、ソース・ノード302と1つ或いは複数のデスティネーション・ノード304との間のコネクション（即ち、論理的な接続関係）を設定、管理する機能と、そのコネクションにて使用される通信プロトコルを選択する機能をもつノードである。

【0065】

ここで、コントローラ300、ソース・ノード302、デスティネーション・ノード304は、夫々独立した別々のノードにおいて機能してもよい。又、コントローラ300とソース・ノード302とが、1つの同じノードにおいて機能してもよい。又、コントローラ300とデスティネーション・ノード304とが、1つの同じノードにおいて機能してもよい。この場合、コントローラ300とソース・ノード302或いはデスティネーション・ノード304との間のトランザ

クションが不要となり、通信手順が簡略化される。

【0066】

本実施例では、コントローラ300、ソース・ノード302、デスティネーション・ノード304の夫々が、独立した別々のノードにおいて機能する場合について説明する。例えば、1394インタフェース14を具備するコンピュータ10が、コントローラ300として機能する。又、1394インタフェース44を具備するDVCR28がソース・ノード302、1394インタフェース62を具備するプリンタ60がデスティネーション・ノード304として機能する。

【0067】

本実施例では、図3に示すように、ソース・ノード302と1つ以上のデスティネーション・ノード304との間に、1つ以上のコネクションを設定することができる。各コネクションは、あるオブジェクトデータの転送要求が生じた場合に、後述するソース・ノード302の出力レジスタ（図5参照）、デスティネーション・ノード304の入力レジスタ（図6参照）に基づいてコントローラ300が設定する。

【0068】

尚、本実施例において、コントローラ300であるコンピュータ10は、デスティネーション・ノード304として動作してもよい。この場合、1つのソース・ノード302と2つのデスティネーション・ノード304との間にコネクションが設定され、オブジェクトデータ308の転送が行われることになる。

【0069】

又、本実施例において、コンピュータ10がコントローラ300として動作する場合について説明したが、コンピュータ10が必ずコントローラ300になる必要はない。DVCR28或いはプリンタ60がコントローラ300として動作してもよい。

【0070】

次に、図5、6を用いて、本実施例のソース・ノード302の有するレジスタ空間とデスティネーション・ノード304の有するレジスタ空間とについて説明する。図5は、ソース・ノード302の有する出力レジスタの構成を説明する図

である。又、図6は、デスティネーション・ノード304の有する入力レジスタの構成を説明する図である。

【0071】

各レジスタ空間は、図4に示すアドレス空間上の所定の領域に用意されており、コントローラ300がソース・ノード302とデスティネーション・ノード304との間にコネクションを設定する場合、該コネクションに第1、第2の通信プロトコルの何れを割り当てるかを選択する場合に使用される。

【0072】

図5(a)は、出力マスターコネクションレジスタ(以下、oMCR)である。oMCRは、主に、ソース・ノード302の有する通信能力について記述しているレジスタ空間(または、メモリ空間)である。

【0073】

oMCRの最上位フィールド501(2ビット)は、max data rateフィールドであり、ソース・ノード302の最大の出力ビットレートを示す。該max data rateフィールドにセットされる値の一例を「表1」に示す。

【0074】

次のフィールド502(1ビット)は、flow control flagであり、本実施例の第1の通信プロトコル(即ち、ブロードキャスト送信を用いたプロトコル)が使用可能か否かを示す。例えば、該フィールドの値が「1」のとき、第1の通信プロトコルが使用可能であることを示す。「0」のときは、第2の通信プロトコルが使用可能である。次のフィールド503(5ビット)は、reservedフィールドであり、将来のために予約されている。

【0075】

【表 1】

「表 1」

Max data rate	最大ビットレート
0 0 ₂	1 0 0 Mbps
0 1 ₂	2 0 0 Mbps
1 0 ₂	4 0 0 Mbps
1 1 ₂	予約

【0076】

次のフィールド504（6ビット）は、extension fieldであり、後述するoutput FIFO sizeフィールドを将来拡張する場合に用いる。次のフィールド505（10ビット）は、output FIFO sizeフィールドであり、データ出力に用いられる先入れ先出し型のバッファメモリ（以下、FIFOメモリ）の容量を示す。

【0077】

次のフィールド506（3ビット）は、reservedフィールドであり、将来のために予約されている。最下位フィールド507（5ビット）は、number of CCRsフィールドであり、後述する出力コネクションコントロールレジスタ（図5（b）参照）の数、即ち、ソース・ノード302に設定することのできるコネクションの数を示す。

【0078】

図5（b）は、出力コネクションコントロールレジスタ（以下、oCCR）である。
oCCRは、ソース・ノード302に設定されたコネクションの1つについて記述するレジスタ空間（又は、メモリ空間）である。ソース・ノード302は、上記oCCRを、唯一つだけ有していても良いし、複数有していても良い。

【0079】

oCCRの最上位フィールド510（2ビット）は、data rateフィールドであり、該oCCRに設定されたコネクションにて用いられる出力ビットレートを示す。「

表 2」に該data rateフィールドにセットされる値の一例を示す。

【0080】

【表 2】

「表 2」

data rate	ビットレート
00 ₂	100 Mbps
01 ₂	200 Mbps
10 ₂	400 Mbps
11 ₂	予約

【0081】

次のフィールド511（1ビット）は、broadcast flagフィールドであり、該oCCRに設定されたコネクションにてどの通信プロトコルが使用されているかを示すフラグである。該broadcast flagフィールドの値が「1」のとき、例えば、第1の通信プロトコルが用いられていることを示す。

【0082】

次のフィールド512（1ビット）は、on-lineフィールドであり、該oCCRが使用されているか否かを示す。該on-lineフィールドの値が「1」のとき、該oCCRは使用されていることを示す。

【0083】

次のフィールド513（18ビット）は、reservedフィールドであり、将来のために予約されている。次のフィールド514（10ビット）は、payload sizeフィールドであり、一回のデータ転送で転送されるデータパケットのデータ量を示す。該データ量はデータパケットのペイロードサイズを示し、パケットヘッダなどの付加データは含まれていない。

【0084】

次のフィールド515（16ビット）は、connection IDフィールドであり、該oCCRに設定されたコネクションを一意に識別するための識別情報、即ちコネク

ションIDが格納される。最後のフィールド516（48ビット）は、offset addressフィールドであり、デスティネーション・ノード304に確保した所定のメモリ空間の先頭アドレスを示す。

【0085】

図6（a）は、入力マスターコネクションレジスタ（以下、iMCR）である。iMCRは、主に、デスティネーション・ノード304の有する通信能力について記述しているレジスタ空間（又は、メモリ空間）である。

【0086】

iMCRの最上位フィールド601（2ビット）は、max data rateフィールドであり、デスティネーション・ノード304の最大の入力ビットレートを示す。該max data rateフィールドにセットされる値は、例えば、上記「表1」と同じで良い。

【0087】

次のフィールド602（1ビット）は、flow control flagフィールドであり、本実施例の第1の通信プロトコル（即ち、ブロードキャスト送信を用いたプロトコル）が使用可能か否かを示す。例えば、該フィールドの値が「1」のとき、第1の通信プロトコルが使用可能であり、「0」のときは、第2の通信プロトコルが使用可能であることを示す。次のフィールド603（5ビット）は、reservedフィールドであり、将来のために予約されている。

【0088】

次のフィールド604（6ビット）は、extension fieldであり、後述するmaximum receive buffer sizeフィールドを将来拡張する場合に用いる。次のフィールド605（10ビット）は、maximum receive buffer sizeフィールドであり、データ入力に用いられる受信バッファの最大メモリを示す。ここで、maximum receive buffer sizeフィールドは、1セグメントの最大データ量を示す。

【0089】

次のフィールド606（3ビット）は、reservedフィールドであり、将来のために予約されている。最下位フィールド607（5ビット）は、number of CCRsフィールドであり、後述する入力コネクションコントロールレジスタ（図6（b

参照)の数、すなわち、デスティネーション・ノード304に設定することのできるコネクションの数を示す。

【0090】

図6(b)は、入力コネクションコントローラレジスタ(以下、iCCR)である。

iCCRは、デスティネーション・ノード304に設定されたコネクションの1つについて記述するレジスタ空間(又は、メモリ空間)である。デスティネーション・ノード304は、上記iCCRを、唯一つだけ有していても良いし、複数有していても良い。

【0091】

iCCRの最上位フィールド610(2ビット)は、data rateフィールドであり、該iCCRに設定されたコネクションにて用いられる入力ビットレートを示す。該data rateフィールドにセットされる値は、例えば、「表2」と同じでよい。

【0092】

次のフィールド611(1ビット)は、broadcast flagフィールドであり、該iCCRに設定されたコネクションにてどの通信プロトコルが使用されているかを示すフラグである。該broadcast flagフィールドの値が「1」のとき、例えば、第1の通信プロトコルが用いられていることを示す。

【0093】

次のフィールド612(1ビット)は、on-lineフィールドであり、該iCCRが使用されているか否かを示す。該on-lineフィールドの値が「1」のとき、該iCCRは使用されていることを示す。次のフィールド613(18ビット)は、reservedフィールドであり、将来のために予約されている。

【0094】

次のフィールド614(10ビット)は、receive buffer sizeフィールドであり、デスティネーション・ノード304に確保したメモリ空間のサイズ、即ち、1セグメントのデータ量を示す。次のフィールド615(16ビット)は、connection IDフィールドであり、該iCCRに設定されたコネクションを一意に識別するための識別情報、即ちコネクションIDが格納される。

【0095】

最後のフィールド616（48ビット）は、offset addressフィールドであり、デスティネーション・ノード304に確保した所定のメモリ空間の先頭アドレスを示す。

【0096】

次に、図5、図6に示す各レジスタの内容を読み出すためのコマンドと、各レジスタに所定の値を書き込むコマンドについて説明する。

各コマンドは、IEEE1394-1995規格のAsynchronous転送方式に準拠するAsynchronousパケットにパケットサイズされ、指定されたノードに転送される。ここで、Asynchronousパケットは、図7に示すように、1Quadlet（4bytes=32bits）を単位とするデータパケットである。

【0097】

本実施例のコントローラ300は、図7に示すREAD MCR status commandを用いて、図5（a）のoMCR或いは図6（a）のiMCRの値を読み出すことができる。

図7において、フィールド701（16bits）は、destination_IDフィールドであり、通信相手のノードIDを指定する。フィールド702（6bits）は、トランザクション・ラベル（tl）フィールドであり、各トランザクション固有のタグを指定する。

【0098】

フィールド703（2bits）は、リトライ（rt）コードであり、パケットがリトライを試みるかどうかを指定する。フィールド704（4bits）は、トランザクションコード（tcode）を示す。tcodeは、パケットのフォーマットや、実行しなければならないトランザクションのタイプを指定する。本実施例では、このフィールドの値を例えば「0001₂」とし、このパケットのデータ・フィールドをdestination_offsetフィールド707の示すメモリ空間に書き込む処理（即ち、ライト・トランザクション）をリクエストする。また、「0005₂」とした場合は、読み出し処理（即ち、リード・トランザクション）をリクエストする。

【0099】

フィールド705 (4 bits) は、プライオリティ (pri) を示し、優先順位を指定する。本実施例では、このフィールドの値は「0000₂」とする。フィールド706 (16 bits) は、source_IDフィールドであり、送信元のノードIDを指定する。

【0100】

フィールド707 (48 bits) は、destination_offsetを示し、destination_IDフィールド701で指定されたノードのアドレス空間を指定する。フィールド708 (16 bits) は、data_lengthを示し、後述するデータフィールドの長さをバイト単位で示す。

【0101】

フィールド709 (16 bits) は、extended_tcodeを示す。本実施例では、このフィールドの値を「0000₂」とする。フィールド710 (32 bits) は、header_CRCであり、上述したフィールド701～709のエラーを検出するためのエラー検出コードを格納する。

【0102】

次に、データ・フィールド (data field) の構成を説明する。本実施例において、データ・フィールドに格納されるデータ量がQualletの倍数に満たない場合、Qualletの倍数となるまで「0」がセットされる。

【0103】

フィールド711 (4 bits) は、Command/Transaction Set (以下、CTSと称する) フィールドである。本実施例では、AV/C Digital Interface Command Set General Specification規格 (以下、AV/C Command Set規格と称する) において定義されているCTSを使用するため、このフィールドには「0₁₆」がセットされる。

【0104】

フィールド712 (4 bits) は、ctype/responseフィールドであり、各種のコマンドやそのコマンドに対するレスポンスを指定する。例えば、status commandを指定する場合、このフィールドには「1₁₆」がセットされる。又、このst

atus commandに対応するレスポンス（即ち、status response）を指定する場合、このフィールドには「C₁₆」がセットされる。

【0105】

フィールド713（5 bits）は、subunit_typeフィールドであり、あるノードの有する仮想的な機能単位（即ち、subunit）を指定する。ここでsubunitとは、AV/C Command Set規格において規定されており、例えば、VTRサブユニット、プリンタサブユニット等がある。

【0106】

フィールド714（3 bits）は、subunit_IDフィールドであり、subunit_typeフィールド713で指定された機能単位が1つ以上ある場合に、それらを識別するためのID番号を指定する。

【0107】

フィールド715（8 bits）は、opcodeフィールドであり、コマンドの種類を指定する。本実施例のREAD MCR commandを指定する場合、このフィールドには、例えば「06₁₆」がセットされる。

【0108】

フィールド716（8 bits）～719（8 bits）は、operand[0]～operand[3]フィールドであり、その内容は、ctype/responceフィールド712、opcodeフィールド715の内容によって異なる。例えば、READ MCR status commandの場合、各フィールドには、「FF₁₆」がセットされる。又、READ MCR status responseの場合、各フィールドには、図5（a）のoMCRの値、或いは図6（b）のiMCRの値がセットされる。

【0109】

フィールド720（8 bits）には、「0」がセットされる。フィールド721（32 bits）は、data_CRCであり、上記のheader_CRCと同様に、フィールド711～720のエラーを検出するためのエラー検出コードを格納する。

【0110】

又、本実施例のコントローラ300は、図8に示すSET CCR control commandを用いて、図5（b）のoCCR或いは図6（b）のiCCRに所定の値を書き込むこと

ができる。

図8において、パケット・ヘッダ (packet header) に含まれるフィールド 701~710は、図7と同様に構成される。従って、以下、SET CCR control commandのデータ・フィールド (data field) の構成について説明する。

【0111】

フィールド801 (4 bits) は、CTSフィールドであり、図7のREAD MCR status commandと同様に、「0₁₆」がセットされる。

【0112】

フィールド802 (4 bits) は、ctype/responseフィールドであり、各種のコマンドやそのコマンドに対するレスポンスを指定する。例えば、control commandを指定する場合、このフィールドには「0₁₆」がセットされる。このcontrol commandが受け付けられたことを示すレスポンスを返す場合、このフィールドにはACCEPTED responseを示す「9₁₆」がセットされる。又、このcontrol commandが受け付けられなかったことを示すレスポンスを返す場合、このフィールドにはREJECTED responseを示す「A₁₆」がセットされる。

【0113】

フィールド803 (5 bits) は、subunit_typeフィールドであり、図7のREAD MCR status commandと同様に、所定のsubunitを指定する。フィールド804 (3 bits) は、subunit_IDフィールドであり、図7のREAD MCR status commandと同様に、所定のsubunitを識別するためのID番号を指定する。

【0114】

フィールド805 (8 bits) は、opcodeフィールドであり、コマンドの種類を指定する。本実施例のSET CCR commandを指定する場合、このフィールドには、例えば「07₁₆」がセットされる。

【0115】

フィールド806 (2 bits) は、data_rateフィールドであり、所望のoCCR (或いはiCCR) に設定されたコネクションが用いる入力ビットレートを示す。該data_rateフィールドにセットされる値は、例えば、「表2」と同じである。

フィールド807 (1 bits) は、broadcast_flagフィールドであり、所望

のoCCR（或いはiCCR）に設定されたコネクションが、どの通信プロトコルを使用するかを示すフラグである。該broadcast_flagフィールドが「1」のとき、例えば、第1の通信プロトコルが設定される。

【0 1 1 6】

フィールド8 0 8（1 b i t s）は、on_lineフィールドであり、所望のoCCR（或いはiCCR）の使用を要求するために「1」をセットする。ここで、所望のoCCR（或いはiCCR）のon_lineフィールド5 1 2（6 1 2）が「1」であれば、該oCCR（或いはiCCR）は既に使用されているため、このSET CCR control commandを受信したノードはREJECTED responseを返す。

【0 1 1 7】

フィールド8 0 9（4 b i t s）は、例えば、固定値「F₁₆」を設定する。フィールド8 1 0（8 b i t s）は、例えば、固定値「F F₁₆」を設定する。フィールド8 1 1（6 b i t s）は、例えば、固定値「3 F₁₆」を設定する。

【0 1 1 8】

フィールド8 1 2（1 0 b i t s）は、oCCRをセットする場合、payload_sizeフィールドであり、iCCRをセットする場合は、receive_buffer_sizeフィールドである。payload_sizeフィールドには、一回のデータパケットで転送可能なデータ量が設定される。又、receive_buffer_sizeフィールドには、1セグメントのデータ量が設定される。

【0 1 1 9】

フィールド8 1 3（8 b i t s）は、図5（b）、図6（b）に示すconnection_IDフィールドの上位8ビットを指定するフィールド、フィールド8 1 4（8 b i t s）は、connection_IDフィールドの下位8ビットを指定するフィールドである。コントローラ3 0 0は、この2つのフィールドにより、所定のconnection_IDを所望のoCCR（或いはiCCR）にセットすることができる。

【0 1 2 0】

フィールド8 1 5（4 8 b i t s）は、offset_addressフィールドであり、デスティネーション・ノード3 0 4に確保した所定のメモリ空間の先頭アドレスを設定する。フィールド8 1 6（8 b i t s）は、CCR_numberフィールドであり、

1つ又は複数個あるoCCR（或いはiCCR）から、所望のoCCR（或いはiCCR）を指定するための番号を設定する。

【0121】

フィールド817（32bits）は、data_CRCであり、フィールド801～816のエラーを検出するためのエラー検出コードを格納する。

【0122】

次に図9を用いて、コントローラ300が、ソース・ノード302とデスティネーション・ノード304との間に論理的な接続関係を設定し、各ノード間を使用する通信プロトコルを設定するまでの手順について説明する。図9において、コントローラ300、ソース・ノード302、デスティネーション・ノード304の三者は、以下に示すネゴシエーションを行う。

【0123】

まず、コントローラ300が、図7に示すREAD MCR status commandパッケージを、ソース・ノード302に対して発行する（901）。ソース・ノード302は、該ノードが有しているoMCRの各フィールド501～507の値をセットしたresponseパッケージを、コントローラ300に返す（901）。

【0124】

次にコントローラ300は、図7に示すREAD MCR status commandパッケージを、デスティネーション・ノード304に対して発行する（902）。デスティネーション・ノード304は、該ノードが有しているiMCRの各フィールド601～607の値をセットしたresponseパッケージを、コントローラ300に返す（902）。

【0125】

上記動作により、ソース・ノード302のoMCRとデスティネーション・ノード304のiMCRとを読み出したコントローラ300は、各レジスタの内容から各ノードの主要な通信能力を判断し、使用可能な通信プロトコルを選択する。

【0126】

例えば、oMCRのflow control flag502とiMCRのflow control flag602の夫々が「1」であった場合、コントローラ300は、ソース・ノード302とデ

ステーション・ノード 304 の夫々が後述する第 1 の通信プロトコル（即ち、ブロードキャストによるデータ転送を行うプロトコル）に対応可能であると判断する。その場合、コントローラ 300 は、第 1 の通信プロトコルの使用を選択する。

【0127】

第 1 の通信プロトコルの選択後、コントローラ 300 は、ソース・ノード 302 とデステーション・ノード 304 との間に論理的な接続関係を設定するために、図 8 に示す SET CCR control command を生成する。

【0128】

ソース・ノード 302 の oCCR（或いは、デステーション・ノード 304 の iCCR）をセットする SET CCR control command において、コントローラ 300 は、broadcast_flag フィールド 807 に「1」を、on_line フィールド 808 に「1」を、connection_ID フィールド 812、813 にコントローラ 300 が管理する使用可能なコネクション ID を、offset_address フィールド 814 にデステーション・ノード 304 の使用可能なデステーション・オフセットを、CCR_number フィールド 815 に未使用の oCCR（又は、iCCR）を指定する CCR 番号を夫々設定する。

【0129】

又、oMCR の flow control flag 502 と iMCR の flow control flag 602 のどちらか一方、或いは両方が「0」であった場合、コントローラ 300 は、ソース・ノード 302 とデステーション・ノード 304 の夫々が後述する第 1 の通信プロトコル（即ち、ブロードキャストによるデータ転送を行うプロトコル）に対応できないと判断する。その場合、コントローラ 300 は、後述する第 2 の通信プロトコルの使用を選択する。

【0130】

第 2 の通信プロトコルの選択後、コントローラ 300 は、ソース・ノード 302 とデステーション・ノード 304 との間に論理的な接続関係を設定するために、図 8 に示す SET CCR control command を生成する。

【0131】

ソース・ノード 302 の oCCR をセットする SET CCR control command において、コントローラ 300 は、SET CCR control command において、broadcast_flag フィールド 807 に「0」を、on_line フィールド 808 に「1」を、payload_size フィールド 811 に一回のデータパケットで転送可能なデータ量を設定する。

又、デスティネーション・ノード 304 の iCCR をセットする SET CCR control command において、receive_buffer_size フィールド 811 に 1 セグメントのデータ量を設定する。

【0132】

更に、コントローラ 300 は、oCCR（又は、iCCR）をセットする SET CCR control command において、offset_address フィールド 814 にデスティネーション・ノード 304 の使用可能なデスティネーション・オフセットを、CCR_number フィールド 815 に未使用の oCCR（又は、iCCR）を指定する CCR 番号を夫々設定する。

【0133】

本実施例において、コントローラ 300 は、receive_buffer_size フィールドに設定する値を、payload_size フィールドに設定する値の整数倍とする。このように構成することにより、データ転送の効率を向上させることができる。

【0134】

次に、コントローラ 300 は、oCCR を設定するために、ソース・ノード 302 に対して、上述の SET CCR control command を発行する（903）。該 SET CCR control command には、コントローラ 300 にて選択された通信プロトコルを示す値や、通信プロトコルに必要な値が含まれている。

【0135】

ソース・ノード 302 の状態が、上記 SET CCR control command に対応可能であった場合には、ソース・ノード 302 は、コントローラ 300 に対して ACCEPTED response を返す（903）。その後、ソース・ノード 302 は、選択された通信プロトコルに基づき、データ転送を開始する。

【0136】

一方、ソース・ノード302の状態が、上記SET CCR control commandに対応できない場合には、ソース・ノード302は、コントローラ300に対してREJECTED responseを返す。この場合、データ転送は開始されず、論理的な接続も開放され、各ノードは始めから再びネゴシエーションを開始する。

【0137】

次に、コントローラ300は、iCCRを開始するために、デスティネーション・ノード304に対して、上述のSET CCR control commandを発行する(904)。該SET CCR control commandには、コントローラ300にて選択された通信プロトコルを示す値や、通信プロトコルに必要な値が含まれている。

【0138】

デスティネーション・ノード304の状態が、上記SET CCR control commandに対応可能であった場合には、デスティネーション・ノード304は、コントローラ300に対してACCEPTED responseを返す(904)。その後、デスティネーション・ノード304は、選択された通信プロトコルに基づき、データ転送を開始する。

【0139】

一方、デスティネーション・ノード304の状態が、上記SET CCR control commandに対応できない場合には、デスティネーション・ノード304は、コントローラ300に対してREJECTED responseを返す。この場合、データ転送は開始されず、論理的な接続も開放され、各ノードは始めから再びネゴシエーションを開始する。

【0140】

以上の手順からなるネゴシエーションにより、コントローラ300は、ソース・ノード302とデスティネーション・ノード304との間の接続と、該接続が使用する通信プロトコルとを設定することができる。尚、図9では、デスティネーション・ノードが1つの場合について説明したが、デスティネーション・ノードが複数個の場合には、上述の処理を各デスティネーション・ノードに対して行えばよい。この場合、コントローラ300は、対象となる全てのノードの通信能力に応じて使用可能な通信プロトコルを選択する。

【0141】

次に、本実施例の第1の通信プロトコルに基づくデータ転送について説明する。

第1の通信プロトコルは、ブロードキャスト通信を利用し、マルチキャスト通信を実現する通信プロトコルである。

【0142】

第1の通信プロトコルにおいて、1つのオブジェクトデータは1つ以上のセグメントに分割され、各セグメントは1つ以上の「Asynchronous broadcast packet」にパケット化され、各Asynchronous broadcast packetは「Asynchronous broadcast transaction」によりブロードキャスト転送される。ここで、Asynchronous broadcast packetとAsynchronous broadcast transactionとは、本実施例の第1の通信プロトコルにおいて規定する全く新規なパケット・フォーマットであり、転送方法である。

【0143】

以下、図10、12を用いて、第1の通信プロトコルに基づく転送手順について詳細に説明する。図10は、1つのソース・ノードと1つのデスティネーション・ノードとの間の転送手順を説明するシーケンスチャートである。又、図12は、1つのソース・ノードと3つのデスティネーション・ノード304との間の転送手順を説明するシーケンスチャートである。

【0144】

上述のネゴシエーションの終了後、コントローラ300は、ソース・ノード302に対してオブジェクトデータ308の転送を開始するように指示する（図10、12の1001）。

転送開始の指示を受けた後、ソース・ノード302は、1つ以上のデスティネーション・ノード304に対して、send request packetをAsynchronous broadcast transactionにより転送する（図10、12の1002）。このsend request packetには、コントローラ300によりセットされたコネクションIDが格納されている。

【0145】

send request packetを受信したデスティネーション・ノード304は、そのsend request packetのもつコネクションIDを、先のネゴシエーションにおいてiCCRにセットされたコネクションIDと比較する。両者が一致する場合、デスティネーション・ノード304はそのsend request packetを受信し、それに対応するack response packetをAsynchronous broadcast transactionにより転送する(図10、12の1003)。ここで、このack response packetには、各デスティネーション・ノード304の確保できる内部バッファのサイズが格納されている。

【0146】

ack response packetの受信後、ソース・ノード302は、Asynchronous broadcast transactionを実行し、1つ以上のセグメントからなるオブジェクトデータ308を順次ブロードキャストする(図10、12の1004~1007)。

【0147】

ここで、図13を用いて、オブジェクトデータ308の転送手順の一例を説明する。図13において、オブジェクトデータ308は、例えばデータサイズが128Kbyteとなる静止画像データである。

【0148】

ソース・ノード302は、send request packetを用いて認識した各デスティネーション・ノード304の受信能力に応じてオブジェクトデータ308を1つ以上のセグメントに分割する。ここで、1セグメントのデータサイズは、各デスティネーション・ノード304の有する内部バッファのサイズによりソース・ノード302が可変的に設定する。例えば、図13のように、オブジェクトデータ308と同じサイズの内部バッファを各デスティネーション・ノードに確保できた場合、ソース・ノード302は、オブジェクトデータ308を1つのセグメントに分割する。

【0149】

次に、ソース・ノード302は、各セグメントを1つ以上のセグメントデータに分割する。図13には、1セグメントを例えば500個のセグメントデータ(1セグメントデータは256byte)に分割する例を示す。

【 0 1 5 0 】

ソース・ノード 3 0 2 は、各セグメントを少なくとも一回の Asynchronous broadcast transaction を用いて順次ブロードキャストする。図 1 3 には、1 セグメントデータを一回の Asynchronous broadcast transaction を用いて転送する例を示す。

【 0 1 5 1 】

1 セグメントの全てを転送した後、次のセグメントがあれば、ソース・ノード 3 0 2 はそのセグメントの転送を準備し、なければ各デスティネーション 3 0 4 とのデータ通信を終了する（図 1 0、1 2 の 1 0 0 8、1 0 0 9）。

【 0 1 5 2 】

次に、図 1 0、1 2 を用いて、コントローラ 3 0 0 の動作について詳細に説明する。

コントローラ 3 0 0 は、ソース・ノード 3 0 2 に対して送信コマンドパケット (transaction command packet) を Asynchronous 転送する（図 1 0、1 2 の 1 0 0 1）。

【 0 1 5 3 】

送信コマンドパケットを受信したソース・ノード 3 0 2 は、コントローラ 3 0 0 から通知されたコネクション ID を用いて初期設定を行い（図 1 0、1 2 の 1 0 0 2、1 0 0 3）、Asynchronous broadcast transaction を実行する（図 1 0、1 2 の 1 0 0 4 ~ 1 0 0 7）。この Asynchronous broadcast transaction により、ソース・ノード 3 0 2 は、1 つ以上のセグメントからなるオブジェクトデータ 3 0 8 を順次転送することができる。

【 0 1 5 4 】

尚、本実施例の第 1 の通信プロトコルにおいて、コントローラ 3 0 0 は、コネクションの接続、非接続を管理する機能を提供するものである。従って、コネクション設定後におけるオブジェクトデータ 3 0 8 の転送は、ソース・ノード 3 0 2 とデスティネーション・ノード 3 0 4 との間のネゴシエーションにより実行される。

【 0 1 5 5 】

一連のAsynchronous broadcast transactionが終了した後、ソース・ノード302は、segment endを示すAsynchronous broadcast packet（以下、segment end packet）をブロードキャストする（図10、12の1008）。

【0156】

コントローラ300は、ソース・ノード302からのsegment end packetを受け取った後、コネクションを解放してデータ転送を終了する。

ここで、segment end packetは、ブロードキャストされるため、そのパケットの内容はデスティネーション・ノード304においても検出することができる。従って、コントローラ300ではなく、デスティネーション・ノード304自体が、ソース・ノード302とのコネクションを解放するように構成してもよい。

【0157】

次に、図10、12を用いて、ソース・ノード302の動作について詳細に説明する。

コントローラ300からのtransaction command packetを受け取ったソース・ノード302は、各デスティネーション・ノード304に対して上述のsend request packetを送出する（図10、12の1002）。

【0158】

ここで、send request packetは、オブジェクトデータ308のAsynchronous broadcast transactionを開始する前に必要な初期情報を得るためのリクエストパケットである。このパケットには、コントローラ300によって指定されたコネクションIDが書き込まれている。

【0159】

デスティネーション・ノード304は、send request packetに対応するレスポンスであるack response packetをブロードキャストする（図10、12の1003）。ここで、ack response packetには、send request packetと同じコネクションIDが格納されている。従って、ソース・ノード302は、受信パケットのコネクションIDを確認することによって、どのコネクションを介して転送されたack response packetであるかを識別することができる。

【0160】

ここで、ack response packetには、各デスティネーション・ノード304に確保されている内部バッファのサイズが格納されている。ack response packetの受信後、ソース・ノード302は、各デスティネーション・ノード304のメモリ空間を共通に指定するデスティネーション・オフセットを用いて、Asynchronous broadcast transactionを開始する。ここで、このデスティネーション・オフセットは、上述のネゴシエーション時に、コントローラ300から指示されたアドレスである。

【0161】

次にソース・ノード302は、最初のAsynchronous broadcast packetを、上述のデスティネーション・オフセットが示すメモリ空間に対して書き込む（図10、12の1004）。このパケットには、コネクションID、セグメントデータのシーケンス番号が格納されている。

【0162】

最初のAsynchronous broadcast packetを送信した後、ソース・ノード302は、デスティネーション・ノード304からのレスポンス・パケットを待機する。デスティネーション・ノード304からは、コネクションIDとシーケンス番号とを格納したレスポンス・パケットがAsynchronous broadcast packetの形式で送出される。このレスポンス・パケットを受け取った後、ソース・ノード302は、シーケンス番号をインクリメントし、次のセグメントデータを含むAsynchronous broadcast packetを転送する（図10、12の1005）。

【0163】

この手順を繰り返して、ソース・ノード302は順次Asynchronous broadcast transactionを行う（図10、12の1006～1007）。デスティネーション・ノード304からのレスポンスを待機する最大の時間はあらかじめ決められており、その時間を過ぎてもレスポンスが帰ってこない場合は、同一シーケンス番号を用いて、同一データを再送する。

【0164】

又、デスティネーション・ノード304から再送を要求するレスポンス・パケットが転送された場合、ソース・ノード302は、指定されたシーケンス番号

のデータを再度ブロードキャストすることもできる。

【0165】

オブジェクトデータ 308 の全てを Asynchronous broadcast transaction を用いて転送した後、ソース・ノード 302 は、segment end packet をブロードキャストし、データ転送を終了する（図 10、12 の 1008）。

【0166】

尚、上述の実施例では、1 セグメントデータの Asynchronous broadcast transaction に伴って必ずレスポンス・パケットを送出するように規定しているが、それに限るものではない。デスティネーション・ノード 304 の有するデータバッファが、複数の連続するセグメントデータによって満たされた後に、デスティネーション・ノード 304 がレスポンス・パケットの送信を行うように構成してもよい。

【0167】

次に、図 10、12 を用いて、デスティネーション・ノード 304 の動作について詳細に説明する。

上述のネゴシエーション後、デスティネーション・ノード 304 は、ソース・ノード 302 からの send request packet を待機する（図 10、12 の 1002）。

【0168】

send request packet を受け取ったデスティネーション・ノード 304 は、そのパケットに書かれているコネクション ID とコントローラから通知されたコネクション ID を確認し、このパケットがソース・ノード 302 からのパケットであるかどうかを判別する。

【0169】

ソース・ノード 302 からの send request packet を受信した後、各デスティネーション・ノード 304 は、コネクション ID、確保できる内部バッファのサイズを書き込んだ ack response packet をブロードキャストする（図 10、12 の 1003）。

【0170】

ソース・ノード 302 から転送された Asynchronous broadcast packet をメモリ空間に書き込んだ後、デスティネーション・ノード 304 は、そのパケットの接続 ID を確認する。そのパケットに含まれる接続 ID が自己の接続 ID と一致する場合、接続 ID とそのパケットに含まれるシーケンス番号とを格納したレスポンス・パケットをブロードキャストする（図 10、12 の 1004～1007）。この場合、受信パケットに含まれるセグメントデータは、内部バッファに格納される。ここで、受信パケットに含まれる接続 ID が自己の接続 ID と異なる場合、デスティネーション・ノード 304 は、その受信パケットを廃棄する。

【0171】

又、デスティネーション・ノード 304 は、受信パケットのシーケンス番号の不整合を検出した場合に、再送要求を示すレスポンス・パケットを送出することもできる。その場合、デスティネーション・ノード 304 は、再送を要求するシーケンス番号をソース・ノード 302 に通知する。

【0172】

全ての Asynchronous broadcast transaction を終了すると、ソース・ノード 302 から segment end packet がブロードキャストされる。このパケットを受信すると、デスティネーション・ノード 304 はデータ転送プロセスを終了する（図 10、12 の 1008）。

【0173】

segment end packet を受信した後、デスティネーション・ノード 304 は、segment end packet を正常に受信したことを示すレスポンス・パケットをブロードキャストする（図 10、12 の 1009）。

【0174】

以上説明したように本実施例の通信システムは、従来の通信方式の不便性を解決することができる。又、リアルタイム性を必要としないデータ転送においても、簡便に高速にデータを転送することができる。

【0175】

又、本実施例では、コントローラ 300 が接続を設定した後、オブジ

エクトデータの転送処理は、コントローラ 300 に制御されることなくソース・ノード 300 と各デスティネーション・ノード 304 との間において実行される。これにより、コントローラ 300 の負荷を減らし、複雑な通信手順を踏むことのない簡単な通信プロトコルを提供することができる。

【0176】

又、本実施例では、デスティネーション・ノード 304 は、各 Asynchronous broadcast transaction に対して必ずレスポンスを返すように構成されている。これにより、リアルタイム性の必要としないデータを確実に転送することのできる通信プロトコルを提供することができる。

【0177】

より確実なデータ転送を実現するためには、バスリセットや何らかの伝送エラーの発生によってデータ転送が中断した場合において、データを欠落させることなく速やかにデータ転送を再開させることが必要である。以下、図 11 を用いて、本実施例の通信プロトコルで規定する再開手順について説明する。

【0178】

例えば、シーケンス番号 i の Asynchronous broadcast packet を受信した後にバスリセットが発生した場合、各ノードは転送処理を中断し、IEEE 1394-1995 規格で定められた手順に従ってバスの初期化、接続構成の認識、ノード ID の設定等を実行する（図 11 の 1010、1011）。

【0179】

バスの再構築が完了した後、各デスティネーション・ノード 304 は、接続 ID とシーケンス番号 i とを格納した再開要求パケット (resend request packet) をブロードキャストする（図 11 の 1012）。

【0180】

Asynchronous broadcast transaction の再開が可能な場合、ソース・ノード 302 は、受信した resend request packet の接続 ID を確認し、この接続 ID を格納した ack response packet をブロードキャストする（図 11 の 1013）。

【0181】

その後、ソース・ノード302は、受信したresend request packetにより要求されたシーケンス番号以降のセグメントデータ、すなわち、シーケンス番号(i+1)で始まるセグメントデータを順次ブロードキャストする(図11の1014)。

【0182】

前述の手順により、コントローラ300、ソース・ノード302、デスティネーション・ノード304は、それぞれのノードIDを考慮することなく、データ転送が中断しても、その後のデータ転送を容易に、かつ、確実に再開することができる。

【0183】

又、前述のように、本実施例では、データ転送が中断した場合にも、コントローラ300の制御手順が簡略化できる効果がある。

【0184】

次に、図14を用いて第1の通信プロトコルで使用するAsynchronous broadcast packetの構成について説明する。Asynchronous broadcast packetは、例えば、1Quadlet (4 bytes = 32 bits)を単位とするデータパケットである。

【0185】

まず、パケット・ヘッダ(packet header)1219の構成を説明する。

図14において、フィールド1201 (16 bits)は、destination_IDを示し、受信先(即ち、デスティネーション・ノード304)のノードIDを示す。本実施例の通信プロトコルでは、オブジェクトデータ308のAsynchronous broadcast transactionを実現するため、このフィールドの値をブロードキャスト用ID(即ち、「FFFF₁₆」)とする。

【0186】

フィールド1202 (6 bits)は、トランザクション・ラベル(tl)フィールドを示し、各トランザクション固有のタグである。

フィールド1203 (2 bits)は、リトライ(rt)コードを示し、パケット

がリトライを試みるかどうかを指定する。

【0187】

フィールド1204 (4 b i t s) は、トランザクションコード(tcode)を示す。tcodeは、パケットのフォーマットや、実行しなければならないトランザクションのタイプを指定する。本実施例では、このフィールドの値を例えば「0001₂」とし、このパケットのデータ・ブロック522をdestination_offsetフィールド1207の示すメモリ空間に書き込む処理（即ち、ライト・トランザクション）をリクエストする。

【0188】

フィールド1205 (4 b i t s) は、プライオリティ(pri)を示し、優先順位を指定する。本実施例では、このフィールドの値は「0000₂」とする。

フィールド1206 (16 b i t s) は、source_IDを示し、送信側（即ち、ソース・ノード302）のノードIDを示す。

【0189】

フィールド1207 (48 b i t s) は、destination_offsetを示し、各デスティネーション・ノード304の有するアドレス空間の下位48 b i t sを共通に指定する。

フィールド1208 (16 b i t s) は、data_lengthを示し、後述するデータフィールドの長さをバイト単位で示す。

【0190】

フィールド1209 (16 b i t s) は、extended_tcodeを示す。本実施例では、このフィールドの値を「0000₂」とする。

フィールド1210 (32 b i t s) は、header_CRCを示し、上述したフィールド1201～1209のエラーを検出するためのエラー検出コードが格納される。

【0191】

次に、データ・ブロック (data block) 1220の構成を説明する。データ・ブロック1220は、ヘッダ・インフォメーション (packet information) 1221とデータ・フィールド (data field) 1

2 2 2 とにより構成される。

【0 1 9 2】

ヘッダ・インフォメーション 1 2 2 1 には、各ノード間の論理的な接続関係（即ち、コネクション）を識別するためのコネクション ID などが格納される。尚、ヘッダ・インフォメーション 1 2 2 1 の構成は、使用目的に応じて異なる。

【0 1 9 3】

又、データ・フィールド 1 2 2 2 は可変長なフィールドであり、上述のセグメントデータが格納される。ここで、データ・フィールド 1 2 2 2 に格納されるセグメントデータがクアドレットの倍数でない場合、クアドレットに満たない分には「0」が詰められる。

【0 1 9 4】

フィールド 1 2 1 1 (1 6 b i t s) は、connection_ID を示し、本実施例のコネクション ID を格納する。本実施例の 1 3 9 4 インタフェースは、このフィールドに格納されたコネクション ID に基づいてソース・ノード 3 0 2 と 1 つ以上のデスティネーション・ノード 3 0 4 との間に設定されたコネクションを識別する。本実施例では、 $2^{16} \times$ (ノード数) のコネクションを確立することが可能となる。これにより、各コネクションの使用する通信帯域の総量が伝送路の容量に達するまで、複数のコネクションを設定することが可能となる。

【0 1 9 5】

フィールド 1 2 1 2 (8 b i t s) は、protocol_type を示し、ヘッダ・インフォメーション 1 2 2 1 に基づく通信手順（即ち、通信プロトコルの種類）を示す。本実施例の通信プロトコルを示す場合、このフィールドの値は例えば「0 1₁₆」となる。

【0 1 9 6】

フィールド 1 2 1 3 (8 b i t s) は、control_flags を示し、本実施例の通信プロトコルの通信手順等を制御する所定の制御データが設定される。本実施例では、このフィールドの最上位ビットを、例えば、再送要求(resend_request)フラグとする。従って、このフィールドの最上位ビットの値が「1」となる場合、本実施例の通信プロトコルに基づく再送要求が生じていることを示す。

【0197】

フィールド1214 (16 bits) は、sequence_numberを示し、特定のコネクションID (フィールド511で指定されたコネクションID) に基づいて転送されるパケットに対して連続的な値 (即ち、シーケンス番号) を設定する。このシーケンス番号によって、デスティネーション・ノード304は、順次 Asynchronous broadcast transactionされるセグメントデータの連続性を監視することができる。不一致が生じた場合、デスティネーション・ノード304は、このシーケンス番号に基づいて再送を要求することもできる。

【0198】

フィールド1215 (16 bits) は、reconfirmation_numberを示す。本実施例においてこのフィールドは、上述の再送要求フラグの値が1である場合にのみ意味を持つ。例えば、上述の再送要求フラグの値が1である場合、このフィールドには、再送を要求するパケットのシーケンス番号が設定される。

【0199】

フィールド1216 (16 bits) は、buffer_sizeを示す。このフィールドには、デスティネーション・ノード304のバッファ・サイズが設定される。

フィールド1217 (16 bits) は、reservedフィールドである。このフィールドは、将来のために予約されている。

【0200】

フィールド1218 (32 bits) は、data_CRCを示し、上記のheader_CRCと同様に、フィールド1211～1217のエラーを検出するためのエラー検出コードが格納される。

【0201】

尚、本実施例の第1の通信プロトコルでは、図14に示すAsynchronous broadcast packetを用いて、上述のAsynchronous broadcast transactionを実現したがそれに限るものではない。例えば、図15に示すAsynchronous stream packetを用いて実現することも可能である。この場合、上述のオブジェクトデータは、1つ以上のセグメントに分割され、各セグメントは1つ以上のAsynchronous stream packetにパケットサイズされる。そして、ソース・ノード302は、各Asynch

ronous stream packetを、Asynchronous転送期間内において順次ブロードキャスト転送する。ここで、Asynchronous stream packetのフォーマットと転送方法とは、IEEE 1394-1995規格の拡張規格であるIEEE 1394. a規格において規定されている。

【0202】

図15において、フィールド1240 (16 bits) は、data_lengthフィールドである。このフィールドには、後述するデータ・フィールド1246の長さをバイト単位にデータが格納される。フィールド1241 (2 bits) は、tagフィールドである。このフィールドには、「00₂」が格納される。

【0203】

フィールド1242 (6 bits) は、channelフィールドである。このフィールドには、このパケットを含む一連のAsynchronous stream packetに割り当てられたチャンネル番号がセットされる。フィールド1243 (4 bits) は、トランザクション・コード (tcode) フィールドであり、Asynchronous stream packetであることを示す「A₁₆」を指定する。

【0204】

フィールド1244 (4 bits) は、synchronization code (sy) フィールドである。このフィールドには、使用するアプリケーションによって決定された制御コードが格納される。フィールド1245 (32 bits) は、header_CRCである。このフィールドには、上述したフィールド1241～1244のエラーを検出するためのエラー検出コードが格納される。

【0205】

フィールド1246 (可変長) は、データ・フィールドである。このフィールドには、図14に示すフィールド1211～1217の値とセグメントデータとが格納される。ここで、フィールド1246に格納されるセグメントデータがクアドレットの倍数でない場合、クアドレットに満たない分には「0」が詰められる。フィールド1247 (32 bits) は、data_CRCである。このフィールドには、上記のheader_CRCと同様に、フィールド1246のエラーを検出するためのエラー検出コードが格納される。

【0206】

次に、本実施例の第2の通信プロトコルについて説明する。

第2の通信プロトコルは、通信相手を特定した通信方式を利用し、ユニキャスト通信を実現する通信プロトコルである。

第2の通信プロトコルにおいて、1つのオブジェクトデータは1つ以上のセグメントに分割され、各セグメントは1つ以上の「Asynchronous write request packet」にパケット化され、各Asynchronous write request packetは「Asynchronous write transaction」により転送される。ここで、Asynchronous write request packetとAsynchronous write transactionとは、IEEE 1394-1995規格において規定するパケット・フォーマットであり、転送方法である。

【0207】

以下、図16を用いて、第2の通信プロトコルに基づく転送手順について詳細に説明する。

上記ネゴシエーション（図9参照）が終わると、最初に、コントローラ300は、デスティネーション・ノード304に、Receive Segment commandを発行する（1301）。受信準備ができ次第、デスティネーション・ノード304は、コントローラ300に、INTERIM responseを発行して待機状態に入る（1302）。

【0208】

次に、コントローラ300は、ソース・ノード302に、Send Segment commandを発行する（1303）。ソース・ノード302は、コントローラ300に、INTERIM responseを発行してデータ送信状態に入る（1304）。

【0209】

データ送信状態に入った後、ソース・ノード302は、Asynchronous write transactionを用いて、所望のオブジェクトデータ308をデスティネーション・ノード304に送信する（1305～1308）。

【0210】

ここで、図13を用いて、オブジェクトデータ308の転送手順の一例を説明する。図13において、オブジェクトデータ308は、例えばデータサイズが1

28 Kbyteとなる静止画像データである。

【0211】

ソース・ノード302は、オブジェクトデータ308を1つ以上のセグメントに分割する。例えば、オブジェクトデータ308と同じサイズの内部バッファをデスティネーション・ノード304に確保できた場合、ソース・ノード302は、オブジェクトデータ308を1つのセグメントに分割する。

【0212】

次に、ソース・ノード302は、コントローラ300から指示されたoCCRのpayload_sizeフィールド514の内容を検出し、各セグメントを1つ以上のセグメントデータに分割する。図13には、1セグメントを500個のセグメントデータ（1セグメントデータは256 byte）に分割する例を示す。ここで、payload_sizeフィールド514の内容は、コントローラ300によりセットされる。

【0213】

次に、ソース・ノード302は、各セグメントを少なくとも一回のAsynchronous write transactionを用いて順次転送する。図13には、1セグメントデータを一回のAsynchronous write transactionを用いて転送する例を示す。

【0214】

1回以上のAsynchronous write transactionにより順次転送されたデータの総量が、receive_buffer_sizeフィールド614に設定された値（即ち、内部バッファの容量）に達した場合、ソース・ノード302は、上記Send Segment commandに対するACCEPTED responseをコントローラ300に対して発行する（1309）。

【0215】

次に、デスティネーション・ノード304は、上記Receive Segment commandに対するACCEPTED responseをコントローラ300に対して発行し、1セグメント分のデータ転送が完了したことを通知する（1310）。次のセグメントがある場合、コントローラ300、ソース・ノード302、及びデスティネーション・ノード304は、上述の手順を繰り返し行う。

【0216】

これにより、ソース・ノード 302 は、Asynchronous write transactionを用いて、所望のオブジェクトデータ 308 をデスティネーション・ノード 304 に対して確実に送信することができる。

【0217】

以上説明したように上述の各実施例では、物理的な接続形態に依存しない論理的な接続関係を IEEE 1394-1995 規格のようなバス型ネットワーク内に構築することができる。

【0218】

又、本実施例では、IEEE 1394-1995 規格に準拠した通信システムにおいて、リアルタイム性は必要とされないが、信頼性が要求される比較的データ量の多いオブジェクトデータ（例えば、静止画像データ、グラフィックスデータ、テキストデータ、ファイルデータ、プログラムデータ等）を、1 以上のセグメントに分割し、各セグメントを 1 つ以上の Asynchronous 転送を用いて順次転送する全く新規な通信プロトコルを提供することができる。

【0219】

又、本実施例では、IEEE 1394-1995 規格に準拠した通信システムにおいて、複数の機器間のデータ通信を、Asynchronous 転送方式によるブロードキャスト通信を用いて実現する全く新規な通信プロトコルを提供することもできる。

【0220】

更に本実施例では、IEEE 1394-1995 規格に準拠した通信システムにおいて、複数の異なる通信プロトコルが存在した場合でも、使用可能な通信プロトコルを自動的に判別し、使い分けることができる。

【0221】

なお、上述の各実施例において説明した通信プロトコル及びそれを実現するために必要な各種の処理動作は、ソフトウェアによって実現することも可能である。

例えば、上述の実施例の機能を実現するためのプログラムコードを記憶した記憶媒体を、各実施例の通信システムを構成する機器の制御部（例えば、図 1 の MP

U12、システムコントローラ50、プリンタコントローラ68)に供給するように構成する。そして、その制御部が、該記憶媒体に格納されたプログラムコードを読み出し、そのプログラムコードに従って各実施例の機能を実現するように通信システム或いは機器自体の動作を制御するように構成しても、上述の実施例を実現することができる。

【0222】

又、上述の実施例の機能を実現するためのプログラムコードを記憶した記憶媒体を、各機器の具備する1394インタフェース14、44、62に供給し、該1394インタフェース14、44、62の動作を制御する制御部(例えば、図2のシリアル・バス・マネージメント206)が、該記憶媒体に記憶されたプログラムコードに従って各実施例の機能を実現するように処理動作を制御するように構成してもよい。

【0223】

この場合、上述の記憶媒体から読み出されたプログラムコード自体が各実施例の機能を実現することになり、そのプログラムコード自体及びそのプログラムコードを制御部に供給するための手段(例えば、記憶媒体自体)は本発明を構成する。

【0224】

かかるプログラムコードを記憶する記憶媒体としては、例えば、フロッピディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、磁気テープ、不揮発性のメモ리카ード、ROMなどを用いることができる。

【0225】

又、上述の記憶媒体から読み出されたプログラムコードが、上述の制御部上で稼動しているOS(オペレーティングシステム)或いは各種のアプリケーションソフト等と共同して、各実施例の機能を実現する場合も本発明に含まれることは言うまでもない。

【0226】

更に、上述の記憶媒体から読み出されたプログラムコードを、上述の制御部に接続された機能拡張ユニットに備わるメモリに格納した後、その機能拡張ユニッ

トに備わる制御部が、該メモリに格納したプログラムコードに従って実際の処理の一部或いは全てを行い、その処理によって各実施例の機能を実現する場合も本発明に含まれることは言うまでもない。

【0227】

尚、本発明はその精神、又は主要な特徴から逸脱することなく、他の様々な形で実施することができる。

例えば、本実施例のコントローラ300は、ソース・ノード302のoMCRとデスティネーション・ノード304のiMCRとを読み出す際、図7に示すREAD MCR status commandを各ノードに発行するように構成したがそれに限るものではない。oMCR、iMCRのアドレスを予め特定のアドレスとなるように構成しておけば、コントローラ300は、各レジスタの内容を、IEEE 1394-1995規格のAsynchronousリード・トランザクションを用いて読み出すこともできる。

【0228】

このように構成した場合、本実施例の1394インタフェースは、図9に示すネゴシエーションの処理の一部を、図2に示すトランザクション・レイヤよりも上位のレイヤに影響を与えることなく実現することができる。これにより、本実施例の1394インタフェースは、アプリケーション・レイヤに対する負荷を少なくすることができ、ハードウェア化が容易になる。

【0229】

又、本実施例のコントローラ300は、ソース・ノード302のoCCRとデスティネーション・ノード304のiCCRに所定の値を書き込む際、図8に示すSET CCR control commandを各ノードに発行するように構成したがそれに限るものではない。oCCR、iCCRのアドレスを予め特定のアドレスとなるように構成しておけば、コントローラ300は、各レジスタの内容を、IEEE 1394-1995規格のAsynchronousリード・ライト・コンペア・スワップ・ロック・トランザクションを用いて書き込むこともできる。

【0230】

このように構成した場合、本実施例の1394インタフェースは、ロック・トランザクションに対する応答により、図9に示すネゴシエーションの処理の一部

を実現することができる。これにより、本実施例の1394インタフェースは、トランザクション・レイヤにおける応答により上述の処理を実現できるため、処理速度をより一層向上されることができる。又、アプリケーション・レイヤに対する負荷を少なくすることができ、ハードウェア化が容易になる。

【0231】

又、本実施例では、コントローラ300が2種類の異なる通信プロトコルを選択する構成及び手順について説明したが、2種類以上の通信プロトコルの中から選択するように構成してもよい。この場合、各ノードは、oMCRのreservedフィールド503或いはiMCRのreservedフィールド506を用いて各通信プロトコルのサポート状態を示すことができる。

【0232】

更に、本実施例では、IEEE1394-1995規格に準拠したネットワークに適用可能な通信プロトコルについて説明したがそれに限るものではない。本実施例の通信プロトコルは、IEEE1394-1995規格のようなバス型ネットワークやバス型ネットワークを仮想的に構成できるネットワークに適用することもできる。

【0233】

従って、前述の各実施例ではあらゆる点で単なる例示に過ぎず、限定的に解釈してはならない。

本発明の範囲は特許請求の範囲によって示すものであって、明細書本文には何等拘束されない。更に、特許請求の範囲の均等範囲に属する変形や変更はすべて本発明の範囲のものである。

【0234】

【発明の効果】

本発明によれば、複数の異なる通信プロトコルが混在する場合でも、システム全体の操作性を著しく損なうことなく、利用可能な通信プロトコルに適応的に選択することができる。

【図面の簡単な説明】

【図1】

本実施例の通信システムの構成例を示すブロック図である。

【図 2】

本実施例の 1 3 9 4 インターフェースの構成を説明する図である。

【図 3】

本実施例の第 1、第 2 の通信プロトコルの基本構成を説明する概念図である。

【図 4】

各ノードの有するアドレス空間を説明する図である。

【図 5】

本実施例のソース・ノードの有する出力レジスタ（oMCRとoCCR）の構成を説明する図である。

【図 6】

本実施例のデスティネーション・ノードの有する入力レジスタ（iMCRとiCCR）の構成を説明する図である。

【図 7】

MCR、iMCRの内容を読み出すためのコマンドの構成を示す図である。

【図 8】

CCR、iCCRに所定の値を書き込むためのコマンドの構成を示す図である。

【図 9】

ソース・ノードとデスティネーション・ノードとの間で使用する通信プロトコルを設定するまでの手順について説明するシーケンスチャートである。

【図 1 0】

本実施例の第 1 の通信プロトコルについて説明するシーケンスチャートで、1 つのソース・ノードと 1 つのデスティネーション・ノードとの間の転送手順を説明する図である。

【図 1 1】

本実施例の第 1 の通信プロトコルについて説明するシーケンスチャートで、通信プロトコルで規定する再開手順について説明する図である。

【図 1 2】

本実施例の第 1 の通信プロトコルについて説明するシーケンスチャートで、1

つのソース・ノードと3つのデスティネーション・ノード304との間の転送手順を説明する図である。

【図13】

第1の通信プロトコルによる通信オブジェクトデータの転送手順を説明する図である。

【図14】

第1の通信プロトコルにて使用されるデータパケットの構成を示す図である。

【図15】

第1の通信プロトコルにて使用されるデータパケットの構成を示す図である。

【図16】

本実施例の第2の通信プロトコルについて説明するシーケンスチャートである。

【符号の説明】

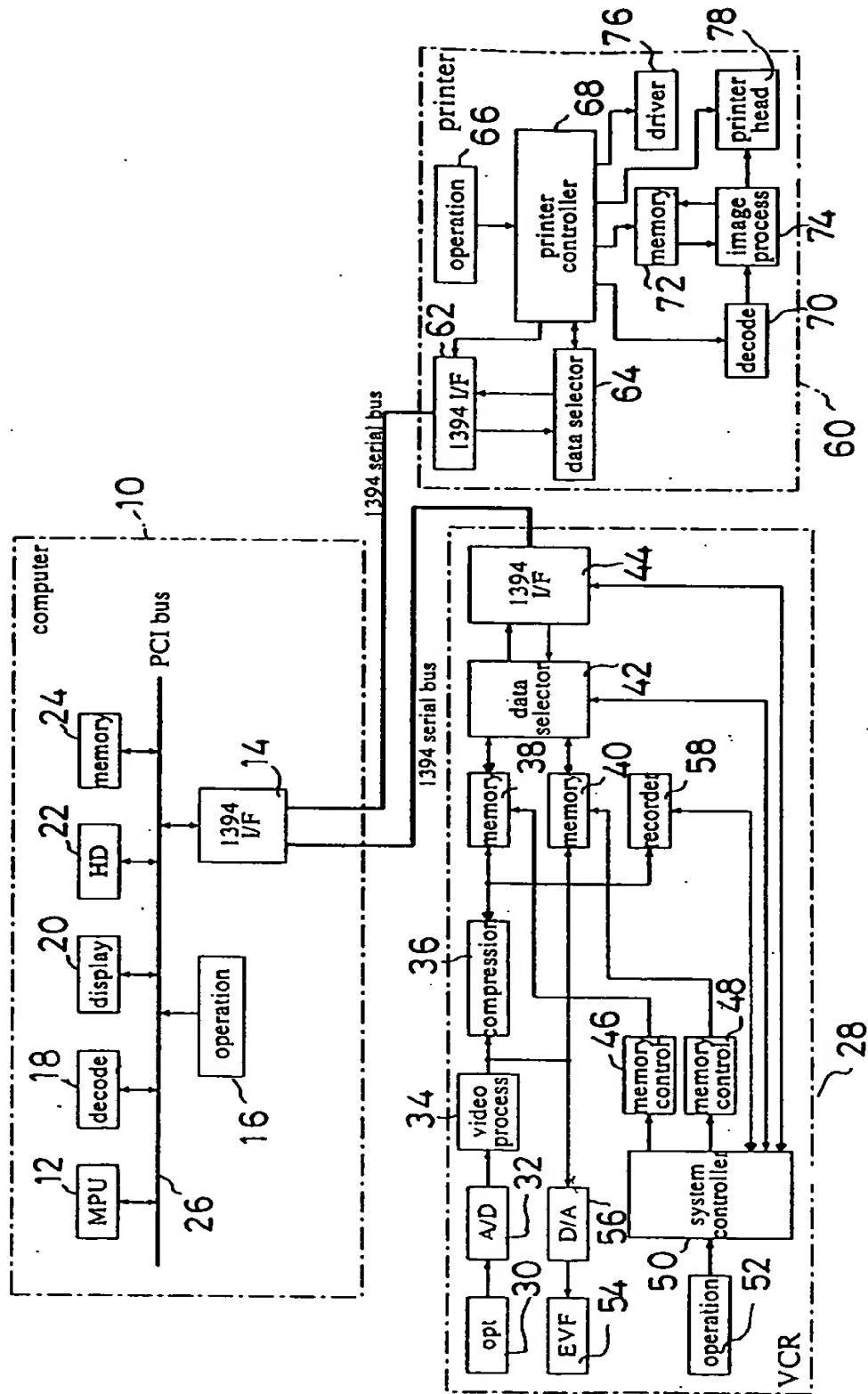
10	コンピュータ
12	演算処理装置
14	1394インターフェース
16	操作部
18	デコーダ
20	表示部
22	ハードディスク
24	内部メモリ
26	内部バス
28	カメラ一体型デジタルビデオレコーダ
30	撮像部
32	A/D変換器
34	画像処理部
36	圧縮／伸長処理部
38、40	メモリ
42	データセクタ

44	1394 インターフェース
46、48	メモリ制御部
50	制御部
52	操作部
54	電子ビューファインダ
56	D/A変換器
58	記録再生部
60	プリンタ
62	1394 インターフェース
64	データセクタ
66	操作部
68	プリンタコントローラ
70	デコーダ
72	内部メモリ
74	画像処理部
76	ドライバ
78	プリンタヘッド
300	コントローラ
302	ソース・ノード
304	デスティネーション・ノード

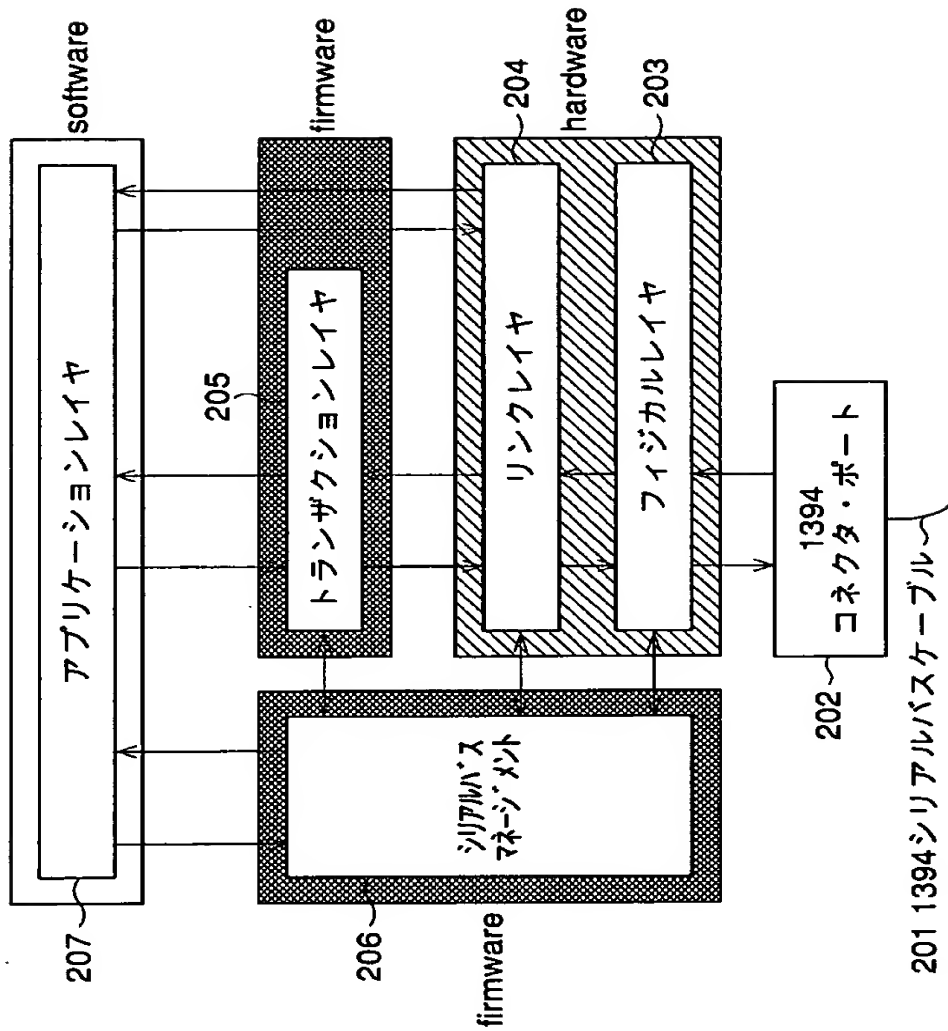
【書類名】

図面

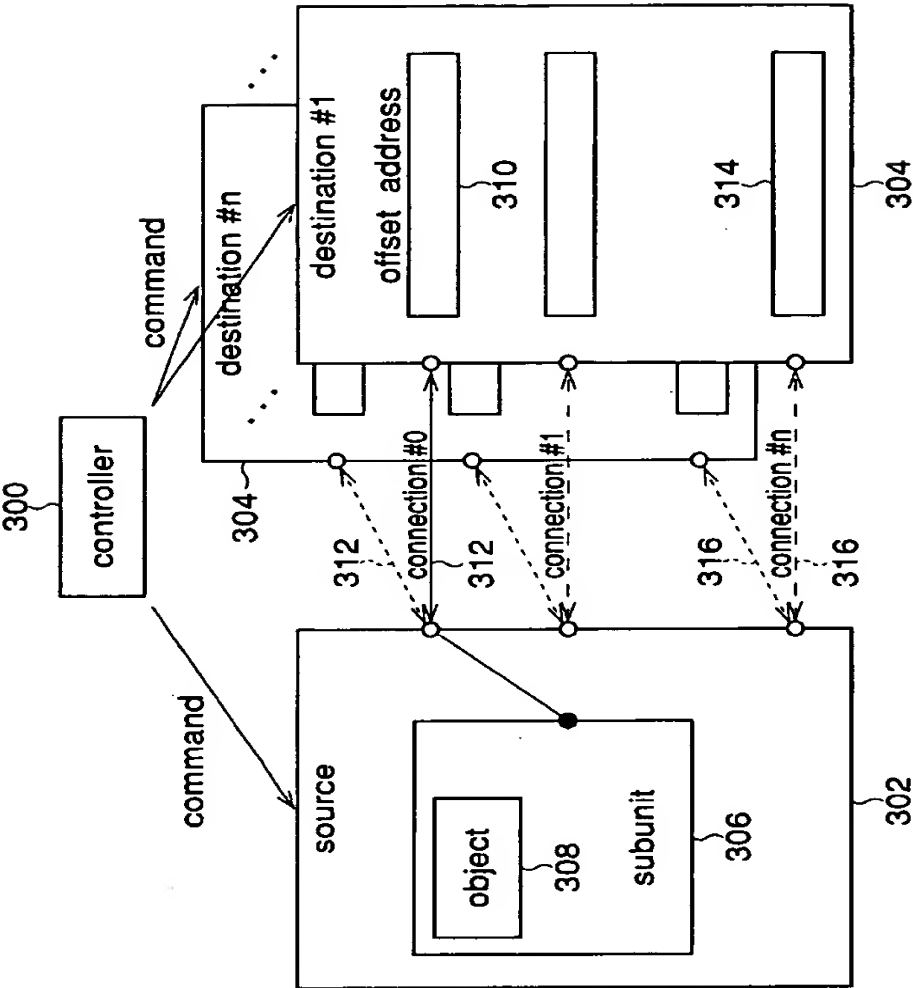
【図 1】



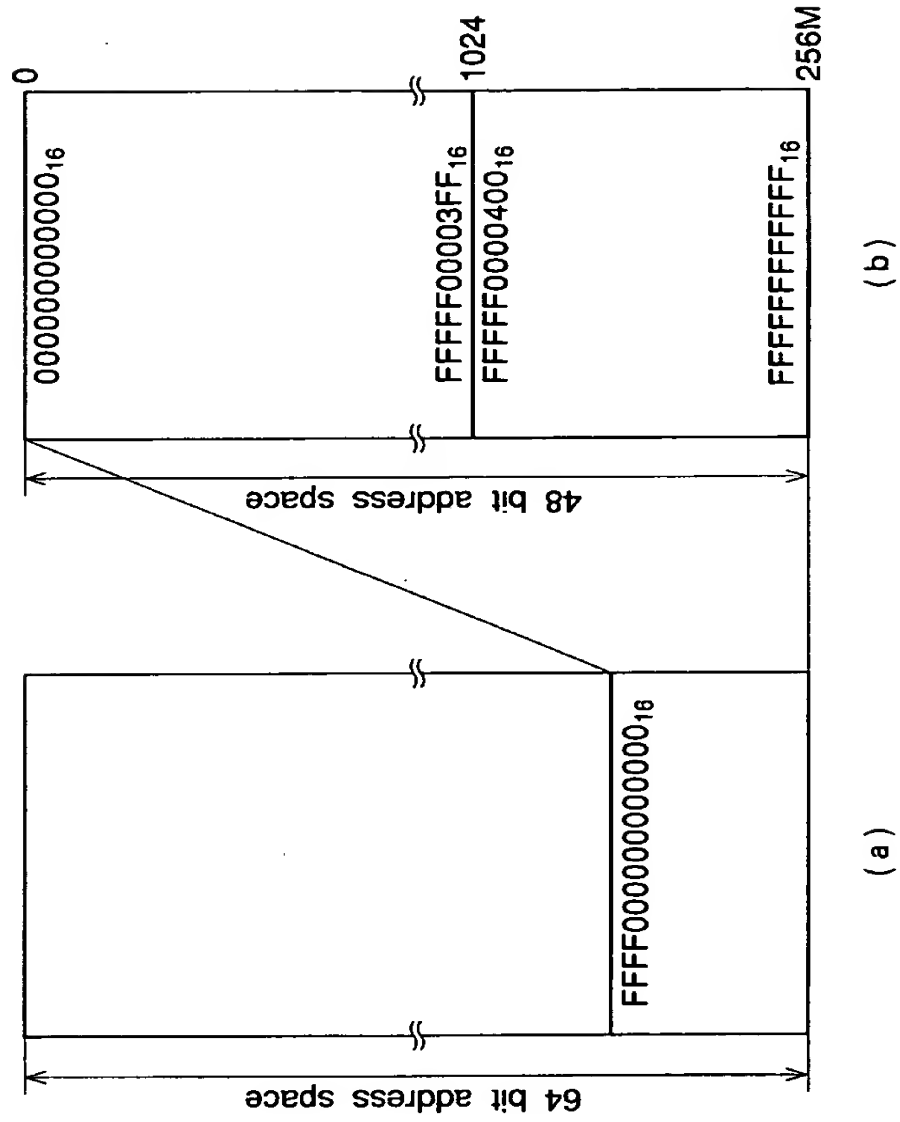
【図 2】



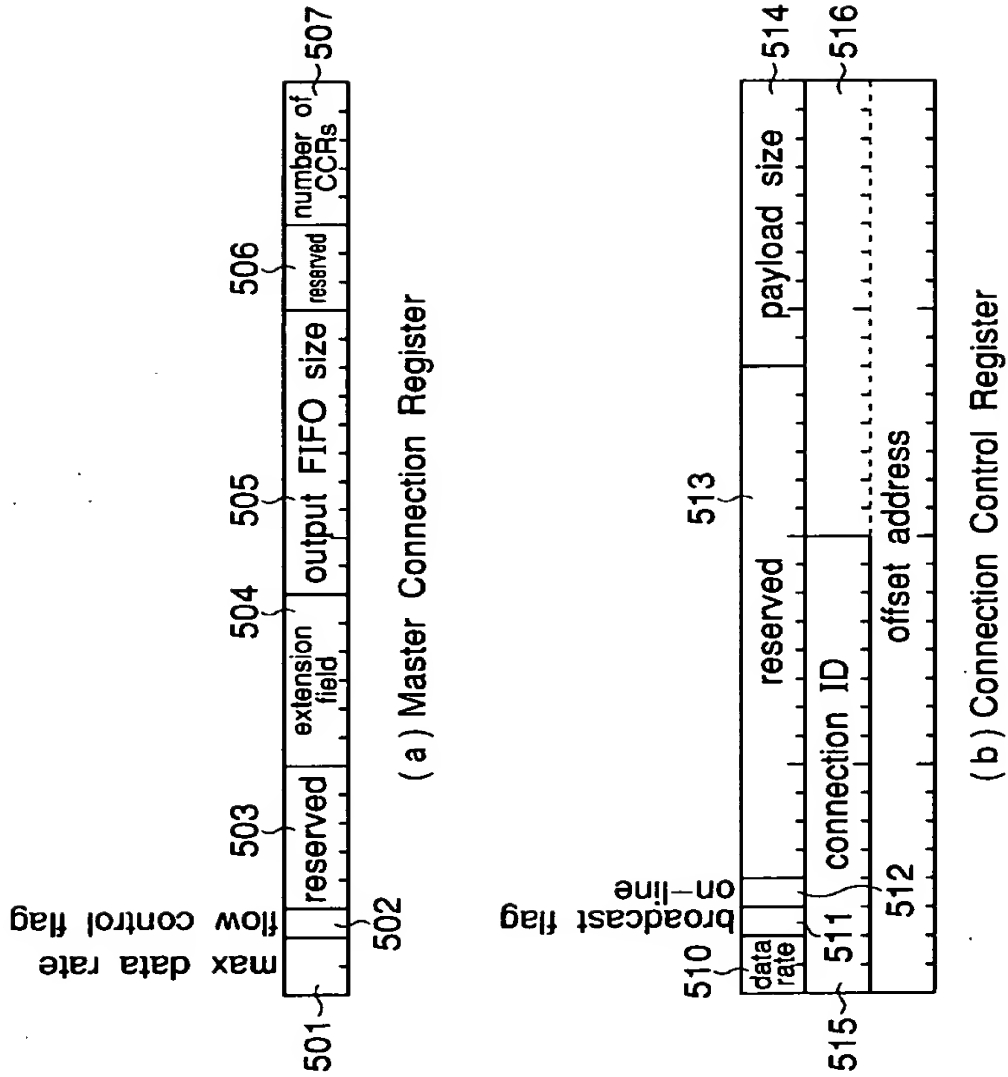
【図 3】



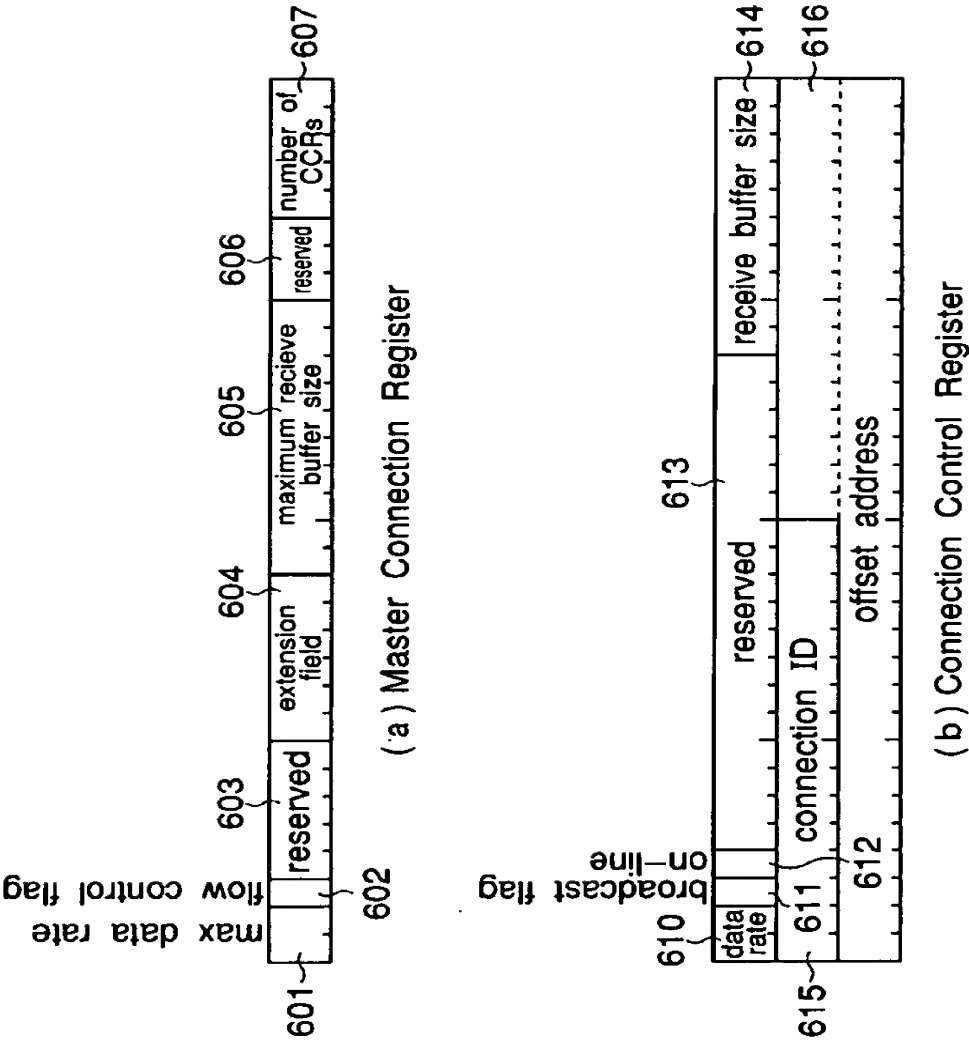
【図 4】



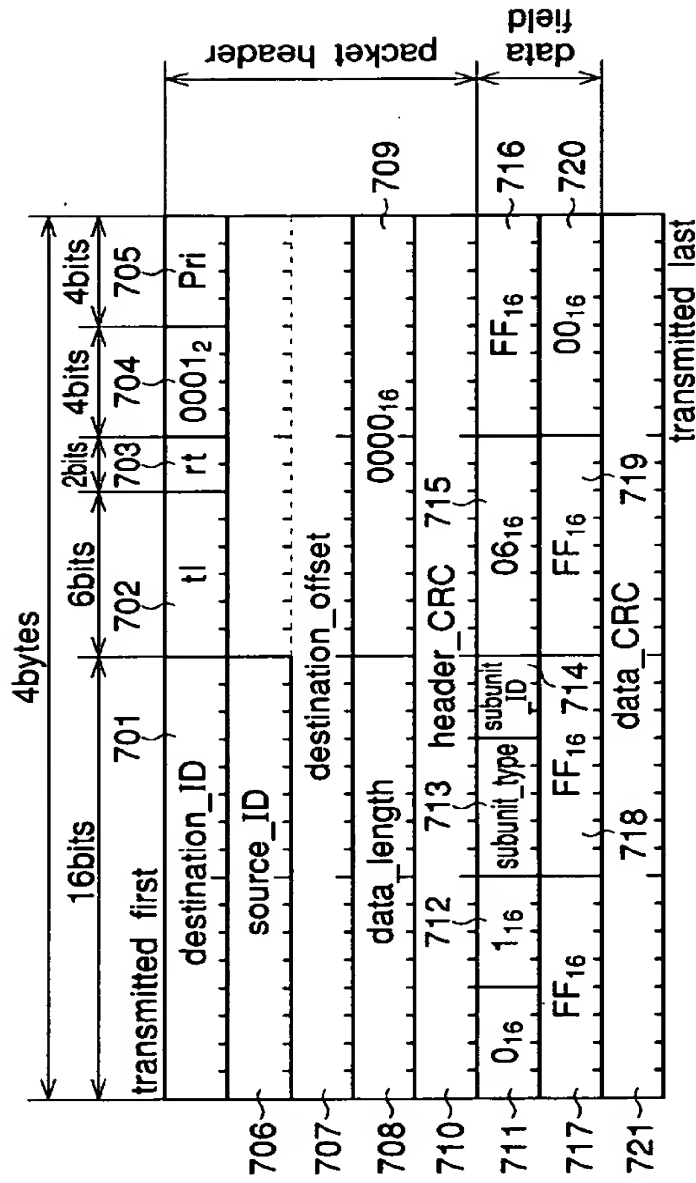
【図 5】



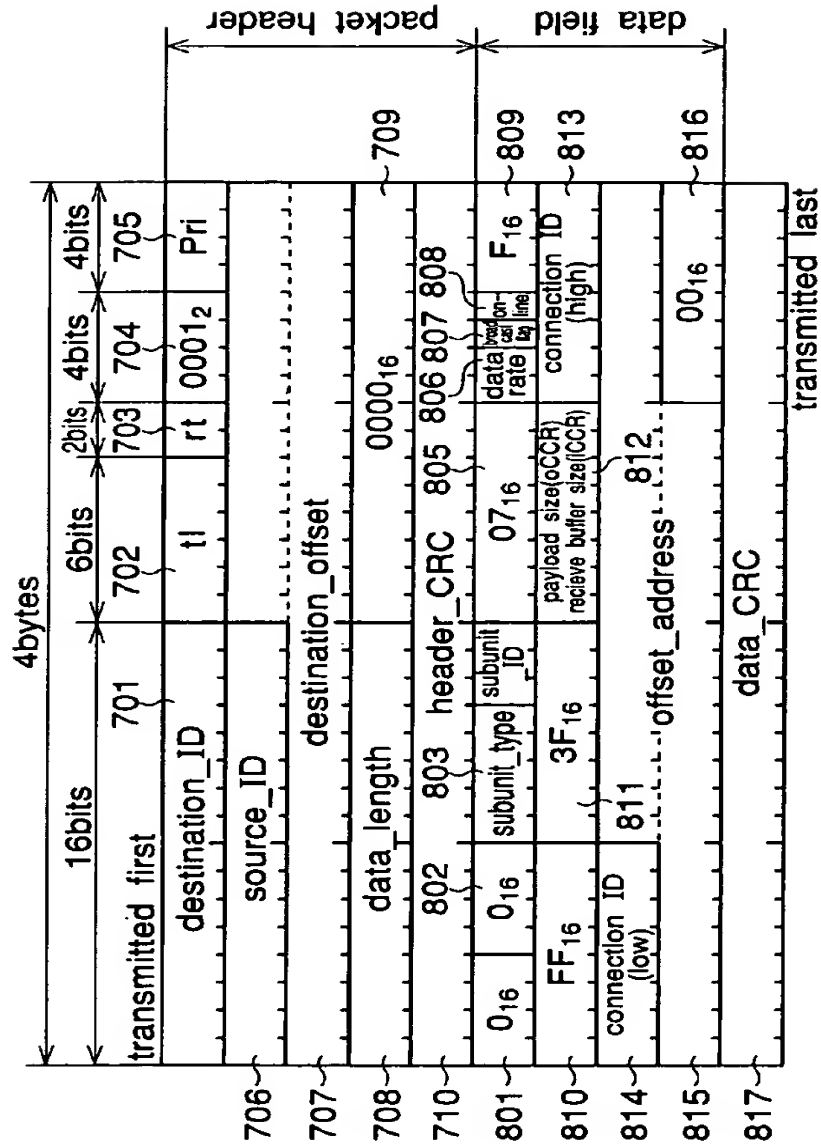
【図 6】



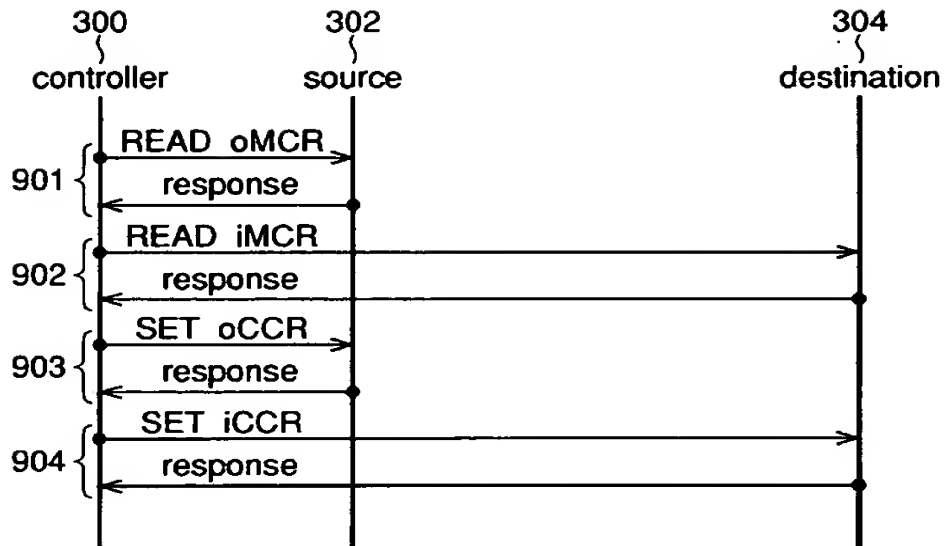
【図 7】



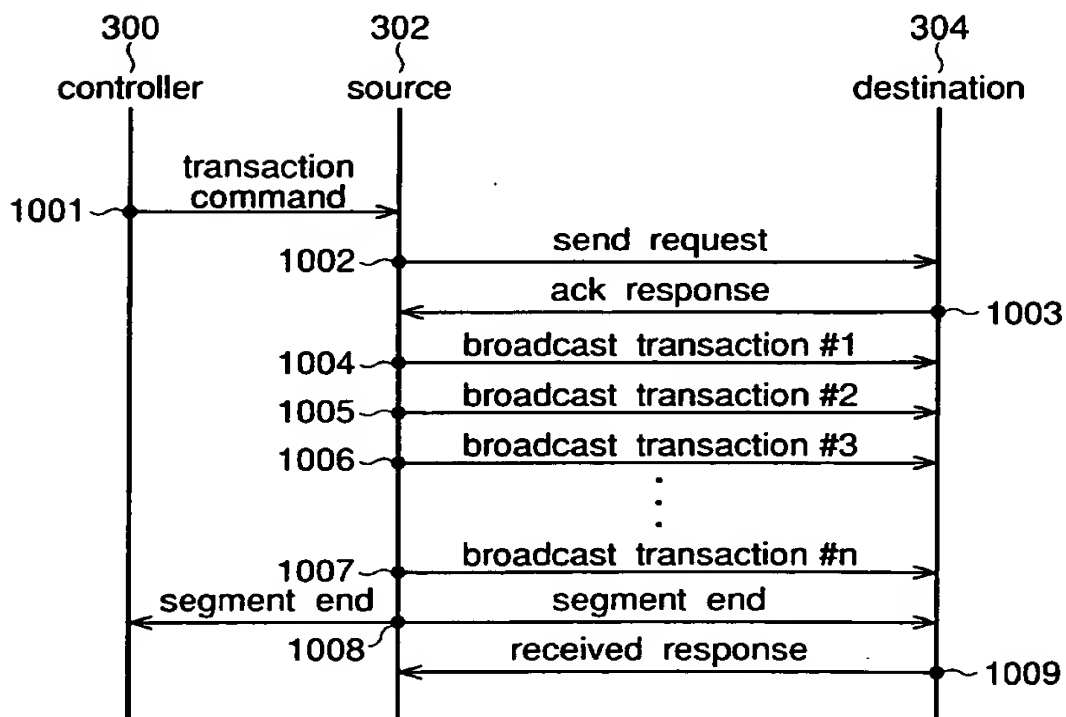
【图 8】



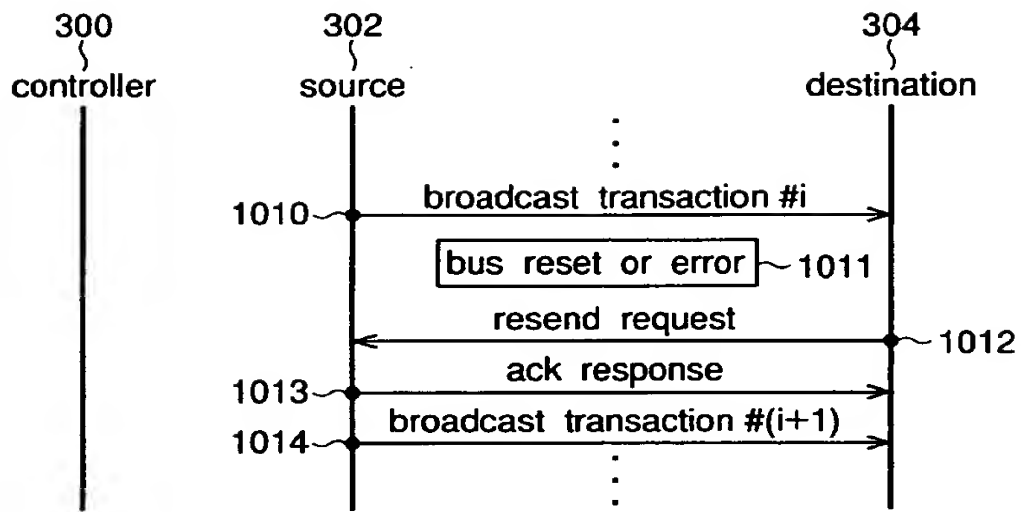
【図 9】



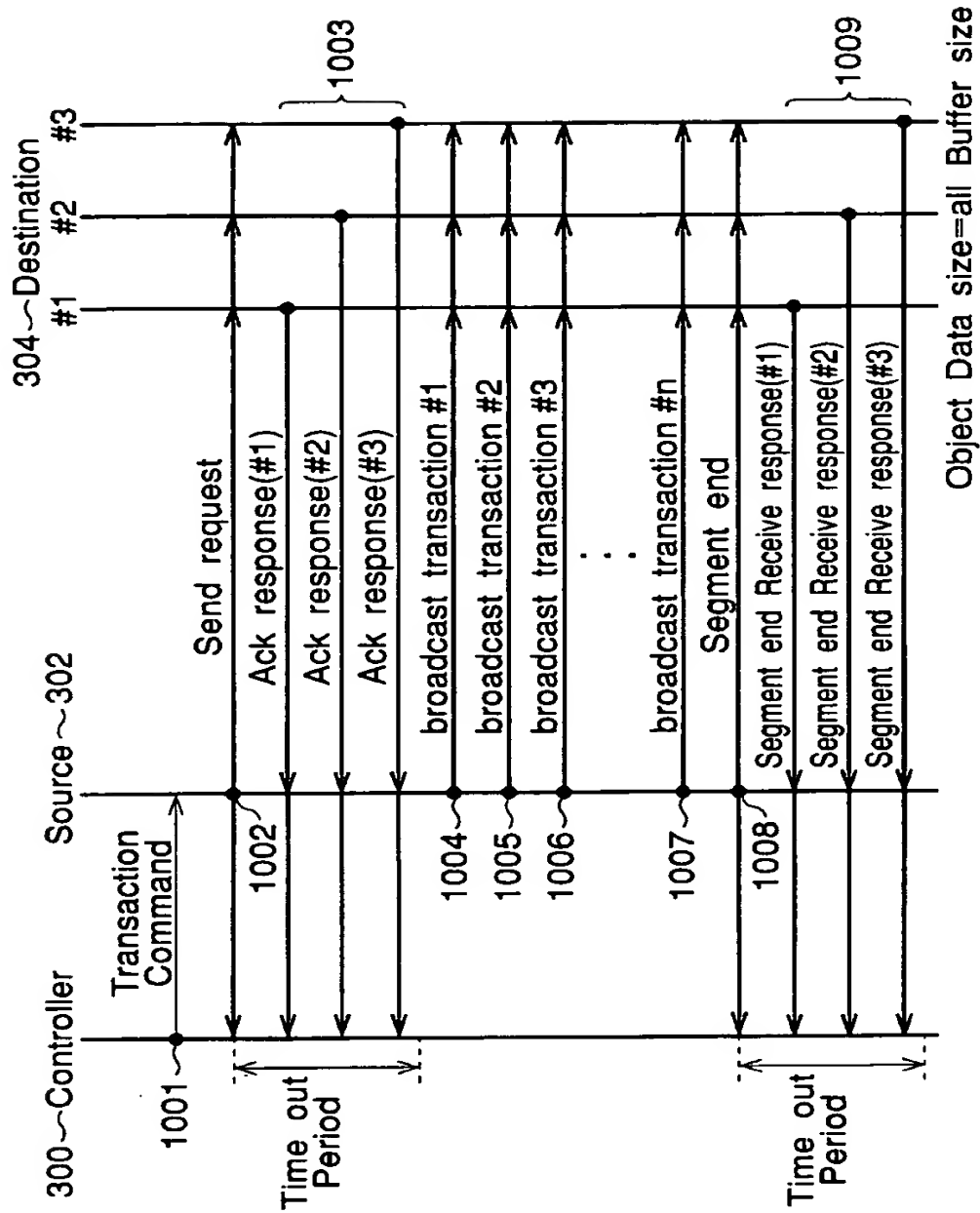
【図 1 0】



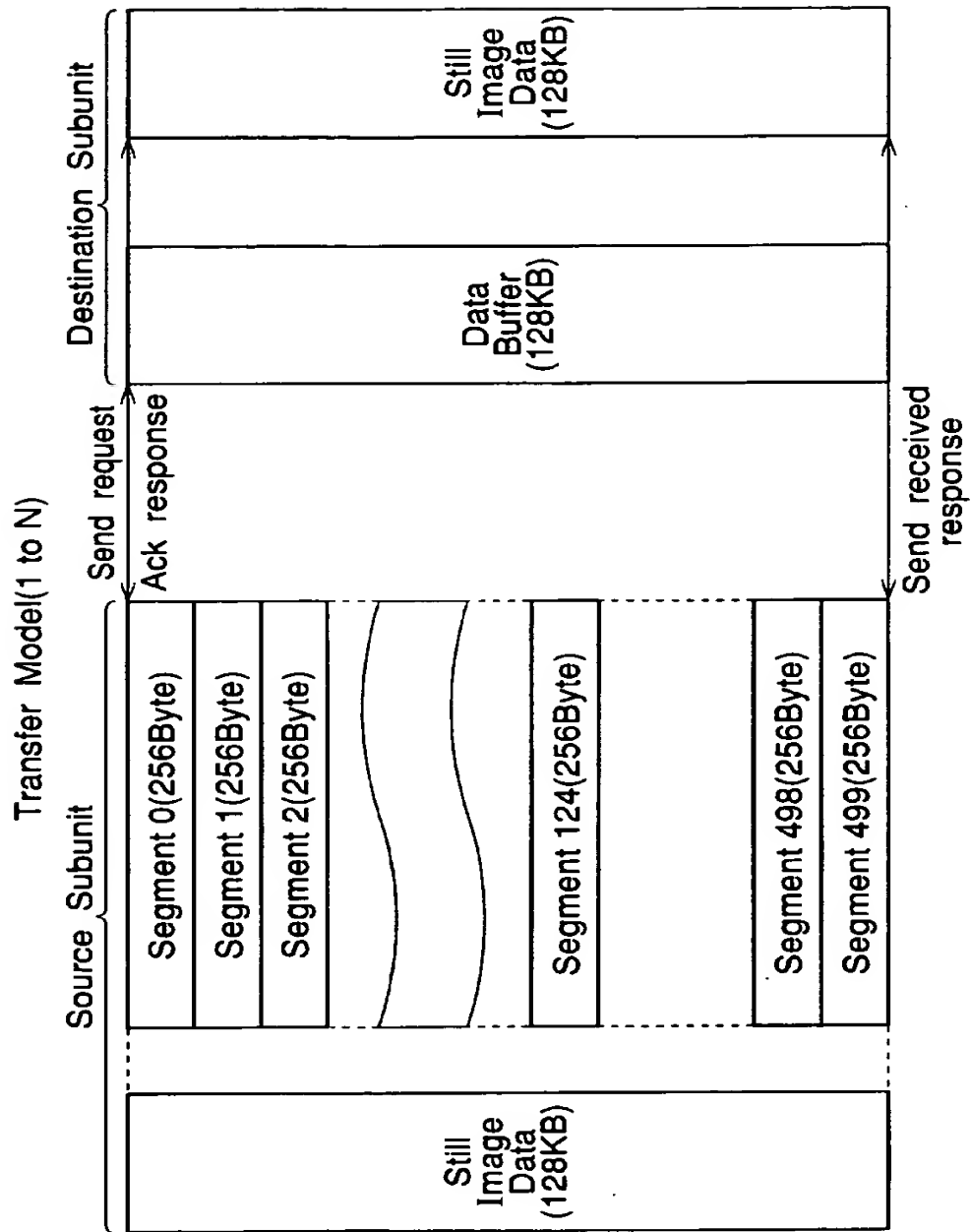
【図 1 1】



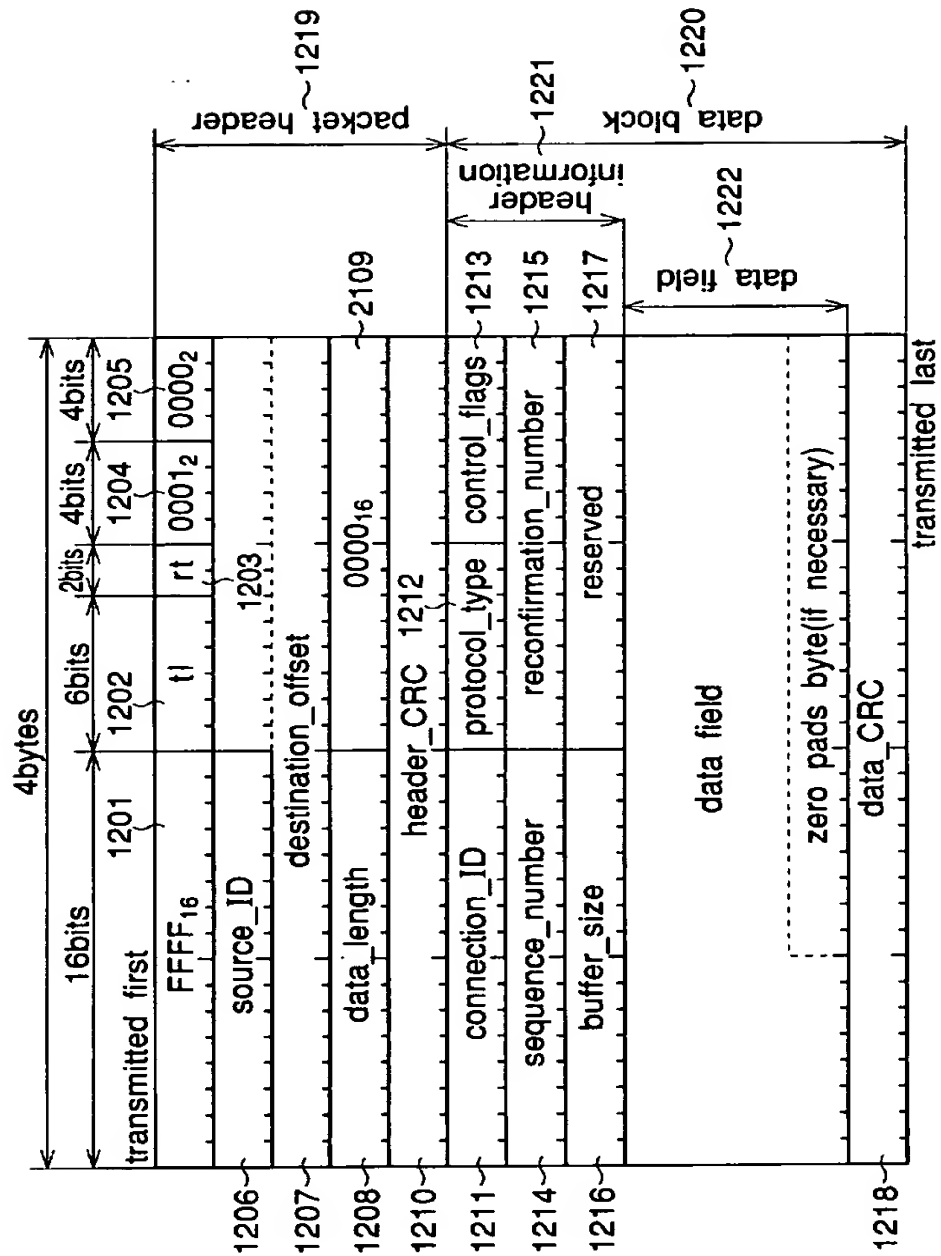
【図 1 2】



【図 1 3】

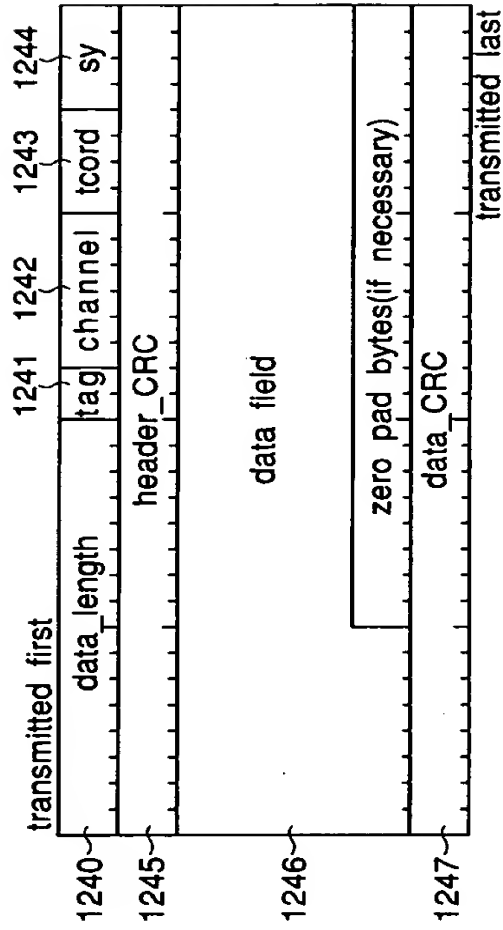


【図 1 4】



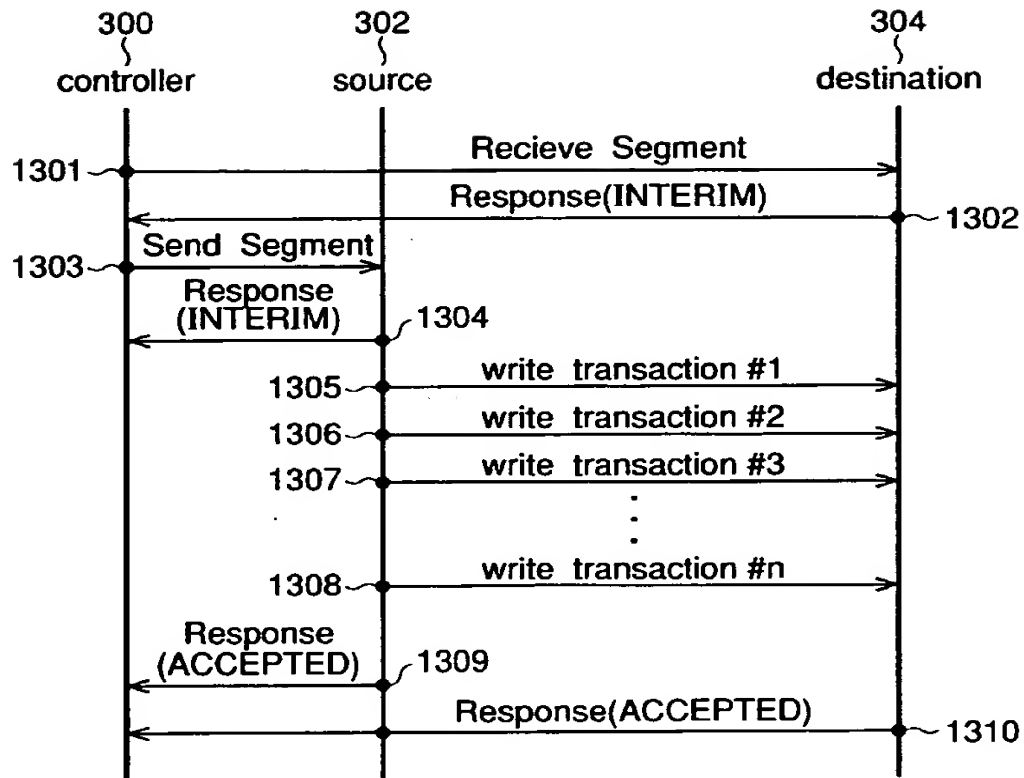
Asynchronous broadcast packet format

【図 1 5】



Asynchronous stream packet format

【図 1 6】



【書類名】 要約書

【要約】

【課題】 複数の異なる通信プロトコルが混在する場合でも、システム全体の操作性を著しく損なうことなく、利用可能な通信プロトコルに適応的に選択することができるようにすることである。

【解決手段】 少なくとも1回の非同期通信を用いて1つ以上のセグメントからなるオブジェクトデータを転送するソース・ノードと、前記ソース・ノードから転送されたオブジェクトデータを受信する1つ以上のデスティネーション・ノードと、前記ソース・ノードと前記1つ以上のデスティネーション・ノードとの間に論理的な接続関係を設定するコントローラとを備え、前記コントローラは、複数の異なる通信プロトコルの中から、前記ソース・ノードと前記デスティネーション・ノードとの間で使用される通信プロトコルを選択する。

【選択図】 図1

出 願 人 履 歴 情 報

識別番号

[000001007]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都大田区下丸子3丁目30番2号

氏 名 キヤノン株式会社